LOAN DOCUMENT

PHOTOGRAPH THIS SHEET

AD-A262 017

DTIC ACCESSION NUMBER

LEVEL

1

INVENTORY

AFOSR-TR-93-0119

DOCUMENT IDENTIFICATION

Dec 92

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

DISTRIBUTION STATEMENT

HANDLE WITH CARE

ACCESSION FOR

NTIS	GRA&I	☒
DTIC	TRAC	☐
UNANNOUNCED		☐
JUSTIFICATION		

BY

DISTRIBUTION/

AVAILABILITY CODES

DISTRIBUTION	AVAILABILITY AND/OR SPECIAL
A-1	

DISTRIBUTION STAMP

DTIC
ELECTE
MAR 1 0 1993
S C D

DATE ACCESSIONED

DTIC QUALITY INSPECTED 1

DATE RETURNED

93 3 10 011

~~93 3 4 069~~

DATE RECEIVED IN DTIC

93-04695

REGISTERED OR CERTIFIED NUMBER

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC

DTIC FORM 70A JUN 90

DOCUMENT PROCESSING SHEET

PREVIOUS EDITIONS MAY BE USED UNTIL STOCK IS EXHAUSTED.

LOAN DOCUMENT

REPORT DOCUMENTATION PAGE

AGENCY USE ONLY (Leave blank)	REPORT DATE	REPORT TYPE AND DATES COVERED
	28 Dec 92	Annual 1 Sep 91 - 31 Aug 92

TITLE AND SUBTITLE

1992 Summer Faculty Research Program (SFRP)
Volumes 1 - 16 V-9

F49620-90-C-0076

AUTHOR(S)

Mr Gary Moore

PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Research & Development Laboratoreis (RDL)
5800 Uplander Way
Culver City CA 90230-6600

AFOSR-TR- 93 0119

SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFOSR/NI
110 Duncan Ave., Suite B115
Bldg 410
Bolling AFB DC 20332-0001
Lt Col Claude Cavender

SUPPLEMENTARY NOTES

DISTRIBUTION/AVAILABILITY STATEMENT

UNLIMITED

ABSTRACT (Maximum 200 words)

The purpose of this program is to develop the basis for cintinuing research of interest to the Air Force at the institution of the faculty member; to stiumlate continuing relations among faculty members and professional peers in the Air Force to enhance the research interests and capabilities of scientific and engineering educators; and to provide follow-on funding for research of particular promise that was started at an Air Force laboratory under the Summer Faculty Research Program.

During the summer of 1992 185 university faculty conducted research at Air Force laboratories for a period of 10 weeks. Each participant provided a report of their research, and these reports are consolidated into this annual report.

SUBJECT TERMS

SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19 SECURITY CLASSIFICATION OF ABSTRACT	
UNCLASSIFIED	**UNCLASSIFIED**	**UNCLASSIFIED**	**UL**

NSN 7540-01-280-5500

PREFACE

This volume is part of a 16-volume set that summarizes the research accomplishments of faculty, graduate student, and high school participants in the 1992 Air Force Office of Scientific Research (AFOSR) Summer Research Program. The current volume, Volume 9 of 16, presents the final research reports of graduate student (GSRP) participants at Rome Laboratory.

Reports presented herein are arranged alphabetically by author and are numbered consecutively -- e.g., 1-1, 1-2, 1-3; 2-1, 2-2, 2-3.

Research reports in the 16-volume set are organized as follows:

VOLUME	TITLE
1	Program Management Report
2	Summer Faculty Research Program Reports: Armstrong Laboratory
3	Summer Faculty Research Program Reports: Phillips Laboratory
4	Summer Faculty Research Program Reports: Rome Laboratory
5A	Summer Faculty Research Program Reports: Wright Laboratory (part one)
5B	Summer Faculty Research Program Reports: Wright Laboratory (part two)
6	Summer Faculty Research Program Reports: Arnold Engineering Development Center; Civil Engineering Laboratory; Frank J. Seiler Research Laboratory; Wilford Hall Medical Center
7	Graduate Student Research Program Reports: Armstrong Laboratory
8	Graduate Student Research Program Reports: Phillips Laboratory
9	Graduate Student Research Program Reports: Rome Laboratory
10	Graduate Student Research Program Reports: Wright Laboratory
11	Graduate Student Research Program Reports: Arnold Engineering Development Center; Civil Engineering Laboratory; Frank J. Seiler Research Laboratory; Wilford Hall Medical Center
12	High School Apprenticeship Program Reports: Armstrong Laboratory
13	High School Apprenticeship Program Reports: Phillips Laboratory
14	High School Apprenticeship Program Reports: Rome Laboratory
15	High School Apprenticeship Program Reports: Wright Laboratory
16	High School Apprenticeship Program Reports: Arnold Engineering Development Center; Civil Engineering Laboratory

1992 GRADUATE RESEARCH REPORTS

Rome Laboratory

Photonic Transversal Filtering for Microwave Systems

Charity A. Carter
Graduate Student
Department of Electrical Engineering
Stevens Institute of Technology
Hoboken, NJ 07030

Final Report for:
Graduate Student Research Program
Rome Laboratory, Photonics Center

Sponsored by:
Air Force Office of Scientific Research

August 1992

Photonic Transversal Filtering for Microwave Systems

Charity A. Carter
Graduate Student
Department of Electrical Engineering
Stevens Institute of Technology

Abstract

Previous papers have shown that it is possible to realize optical processors that can be used in wideband transmit and receive systems which would otherwise be restricted to narrowband use [1,2,3]. The continuously variable time delay supplied by these processors makes them ideal for use in systems, such as phased arrays, which employ transversal filtering. This report discusses the implementation of acousto-optic based and fiber optic based optical processors in these systems. In addition, non-uniform sampling, arising from the availability of a continuously variable delay, is presented. Finally, it is shown that a wavelet transform can be used to perform the correlation required for a matched filter used to extract information from a radar signal.

Photonic Transversal Filtering for Microwave Systems

Charity A. Carter

Introduction

Most beamforming methods used for phased array antenna systems are limited to narrowband operation due to a lack of availability of a continuously variable time delay. If wideband operation is desired, variable delay is necessary since ,without it, the beamforming error known as squint will result. However, by making use of photonic methods, continuously variable time delays that can be utilized in both wideband phased array transmit and receive systems have been constructed [1,3] . In the transmit mode, an acousto-optic cell addressed by a deformable mirror device is used in the construction of the delay line system. A segmented mirror device and a fiber optic delay line array are utilized in the receive system.

Transversal filters, which are used in many signal processing filter applications, employ amplitude weighting and time delay in their architecture. Given the existence of continuously variable time delay, it would be possible to incorporate this into a system to yield a filter capable of continuous reconfigurability [2] .

Matched filters are used in radar systems to minimize the effects of noise in the process of signal detection [5]. These filters make use of a correlation integral in order to distinguish signal information from noise. Due to the availability of a continuously variable time delay, a phased array radar system can be used for wideband operation. It has been shown that, for a wideband signal, the continuous time wavelet transform becomes a "wideband cross-ambiguity function". Therefore a wavelet transform can be implemented in wideband radar systems to carry out the matched filter operation [4].

This report focuses on the implementation of continuously variable time delay for wideband use of microwave systems. This investigation formed the basis of the summer research project. Further research will be carried out in these areas especially that of the implementation and optimization of the variable time delay in systems utilizing transversal filter architectures.

Acousto-Optic Based Optical Processor

In order to realize a continuously variable time delay, an acousto-optic cell operating in the Bragg regime has been utilized in an optical heterodyne configuration [1]. The heterodyne system forms the basis for obtaining phase information while the AO cell acts as a frequency shifter of the laser light source.

The process of optical heterodyning involves splitting the output of a laser into two paths. In one of the paths, the beam is frequency shifted and then optically phase shifted. The other beam provides a phase reference and is referred to as the local oscillator beam. The two beams are then summed and input to a photodetector that responds to the time-average intensity of the light. The output produced contains an RF frequency component equal to the optical beat frequency and an RF phase shift equal angle the optical phase shift. This optical frequency shift varies linearly with frequency as required for a true time delay.

Of fundamental importance in achieving continuously variable time delay, is the use of an acousto-optic cell. The AO cell provides frequency translation and spatial dispersion of the optical frequencies. In addition, the AO cell acts as the energy storage device necessary for any physical delay. The acoustic wave, which propagates along the cell, is produced by a piezoelectric transducer whose input is an RF signal. The RF time delay introduced is determined by the time required for the acoustic wave to propagate a portion of the laser beam width through the AO cell.

Fiber Based Optical Processor

The above system, using a single optically tapped AO cell, functions efficiently in the transmit mode of operation when used in the formation of an electromagnetic radiation pattern for phased array antennas. However, in the receive mode, the non-reciprocal character of acousto-optic modulators would require the use of one AO cell per antenna element. If weight and power constraints exist, this approach would be impractical. It has been shown that an optical processor for the control of broadband phased array receive systems can be constructed based on the use of a spatial light modulator known as a segmented mirror device (SMD) [3]. This device consists of an NxN array of controllable mirror elements which serve to steer light into an array of fiber delay lines. Each fiber in the array is of a different length , therefore the light traveling through

one fiber experiences a delay different from light passing through any other fiber. The output of these fibers can then be summed and fed into a detector to yield a system with variable delay. This receive configuration, making use of a SMD, is well suited for applications involving large arrays. Due to its wide electrical bandwidth characteristic, it overcomes the dispersive beampointing error known as squint, which is associated with narrowband systems.

Transversal Filtering

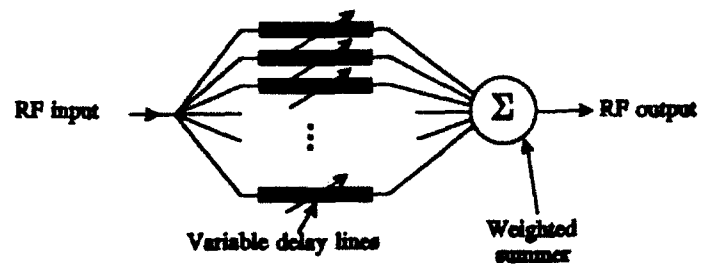
Transversal filtering, which makes use of tapped-delay lines, is a powerful signal processing technique. In general, the output of an adaptive (reconfigurable) transversal filter consists of the sum of weighted and delayed versions of an input signal [6]. Therefore, for an N element transversal filter, the equation for the output, y(t), is given by:

$$y(t)=\sum_{i=1}^{N} a_i x(t-T_i) \qquad (1)$$

where x(t) represents the input signal, T_i represents the time delays and a_i represents the amplitude weight values.

In a typical adaptive filter, the output is compared to the desired response signal. An error exists if there is any difference between the two signals. However, since the filter is adaptable, the weights a_i can be adjusted to reduce the error present in y(t). In most cases, the optimization of the weights is carried out in a manner resulting in the minimization of the mean square value or average power of the error signal. The mean square error is a quadratic function of the amplitude values a_i, which are often collectively referred to as the weight vector. A plot of the mean square error versus the amplitude weight values gives a " bowl-shaped" surface commonly termed the performance surface. By finding the minimum of the performance surface, which is typically done by gradient methods, the optimal weight values can be determined. In traditional adaptive filters, the values of T_i are uniform and do not aid in the adaptation process [6].

The previous adaptive transversal filter configuration contains only one degree of freedom,



TRANSVERSAL FILTER

namely the a_i values. However, when a continuously variable delay line is employed to construct the delay elements, it becomes possible to design a system in which the individual delays may be set for different values. Therefore, a second degree of freedom exists. The adaptation process can then be carried out using the a_i and T_i values which may be adjusted to aid in the optimization of the output and reduce the error present. In order to efficiently make use of this capability, it would be necessary to develop an algorithm that will carry out the optimization of the adjustable parameters.This topic will be the subject of further research.

The availability of variable time delay can be utilized in systems using non-uniform sampling. This will be discussed below.

Impulse Response & Frequency Response Synthesis

The impulse response and frequency response of a transversal filter with a single input x(t) can be found by taking the Fourier transform of the filter output as given by equation (1) [6]. This yields

$$Y(\omega)=\sum_{i=1}^{N} a_i \exp(-j\omega T_i)X(\omega) \qquad (2)$$

By comparing equation (2) to the frequency domain version of the convolution property given by

$$Y(\omega)=H(\omega)X(\omega) \tag{3}$$

it can be seen that the frequency response (transfer function) of a transversal filter is

$$H(\omega)=\sum_{i=1}^{N} a_i \exp(-j\omega T_i) \tag{4}$$

It follows that, by taking the inverse Fourier transform of the frequency response, the filter impulse response is

$$h(t)=\sum_{i=1}^{N} a_i \delta(t-T_i) \tag{5}$$

where $\delta(t)$ is the Dirac delta function.

Non-Uniform Sampling

According to the sampling theorem, it is possible to reconstruct a signal from its samples provided that the function is bandlimited and that the sampling frequency is greater than twice the highest frequency of the signal [7]. This theorem was developed for the case when x(t) is sampled by an impulse train at intervals spaced by T to produce a sampled function of the form

$$x_s(t)=\sum_{n=-\infty}^{+\infty} x(nT)\delta(t-nT) \tag{6}$$

In the equation for $x_s(t)$, the signal x(t) is sampled at uniform intervals, that is at integer multiples of T.

Given that it is possible to construct continuously variable delay lines, a signal can be sampled at non-uniform intervals. It has been shown that, for the case of non-uniform sampling, a result similar to that of the sampling theorem can be obtained [8]. It was concluded that the sampled waveform $v_s(t)$ can be expressed as a sum of exponentials multiplied by the original signal v(t) and an amplitude factor. The first term in the exponential expansion has the same form as the original waveform except for an amplitude distortion. If the frequency band occupied by this term does not overlap the frequency bands of the higher order terms, low-pass filtering will separate it from those terms. In this manner, the original waveform can be recovered even though non-uniform sampling was used.

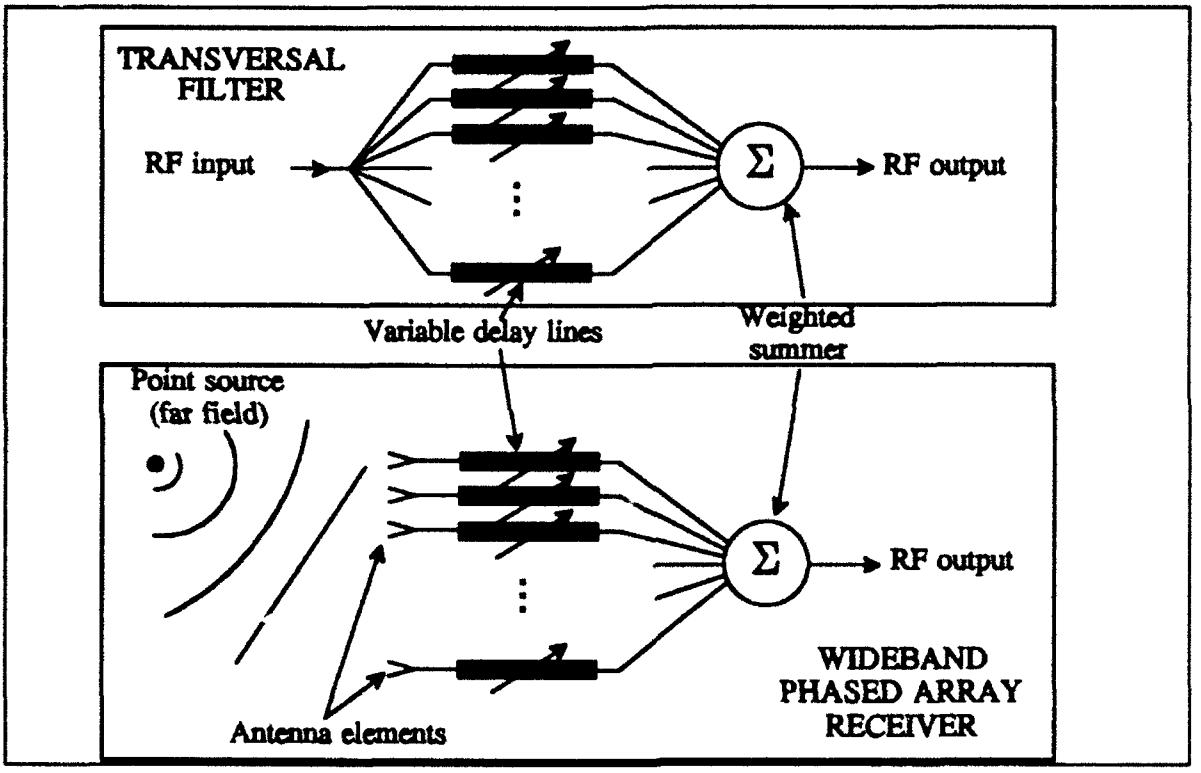
In addition, non-uniform sampling has found extensive use in bandpass filtering [8]. It has been shown that it is not possible to produce an asymmetric passband centered at frequency f_0 from a single filter containing samples uniformly spaced at intervals of $1/2f_0$ [9].

Phased Array Antenna Analogy

In a phased array antenna system, the desired radiation pattern is obtained by controlling the relative amplitude and phase of the signals applied to the individual radiating elements [5]. The combined effect of all the elements determines the shape and direction of the far field electromagnetic beam that has been formed. By inserting an appropriate phase shift at each element of the array, the main beam of the antenna pattern may be made to point in a given direction. One method of introducing a phase shift involves delaying the signal sent to each element of the array as the signal propagates across the face of the array. A relative delay, and therefore phase shift, will exist between the array elements. Phase shift can be introduced by utilizing devices such as diode and ferrite phase shifters. However, these do not provide sufficient bandwidth if wideband operation is required.

In a traditional phased array system, a continuously variable delay is not available. This has restricted the use of phased array antenna systems to narrowband applications due to the presence of squint [1]. Implementing the continuously variable delay line previously described, allows the utilization of phased array systems for wideband applications.

If a phased array system is used in the receive mode, there exists a similarity to a transversal filter system [3]. When information from a far field point source is received by a phased array system, each antenna element will receive a delayed version of that signal. For the phased array antenna, varying the delay enables steering the beam in different directions. Therefore, in a phased array receive system, varying the delays allows information from different directions to be detected.



Matched Filter

In order to determine the range and velocity of a target, a radar system first emits a known signal. After being reflected by the target,the received signal contains the desired information. However, noise is also received along with the signal. A network whose frequency-response maximizes the output peak-signal-to-noise ratio is required to detect the signal in the presence of noise. One such network is the matched filter and it is used in the design of most radar receivers [5].

The frequency response of a matched filter is

$$H(\omega)=kS^{*}(\omega)\exp(-j\omega t_1) \tag{7}$$

where $S^{*}(\omega)$ is the complex conjugate of the Fourier transform of the received signal s(t), k is a constant equal to the maximum filter gain(generally taken to be one), and t_1 is a fixed value of time at which the signal is observed to be a maximum. It can be seen that the amplitude spectrum of the matched filter is the same as that of the signal and the phase spectrum of the matched filter is the negative of that of the signal plus a phase shift proportional to frequency. By taking the inverse transform of the frequency response, the impulse response can be seen to be

$$h(t)=ks(t_1-t) \tag{8}$$

The output of a matched filter is not an exact replica of the input signal. However, the output is proportional to the input signal cross-correlated with a replica of the transmitted signal, except for the time delay t_1. The cross-correlation of two signals is defined as

$$R(t)=\int_{-\infty}^{\infty} y(\lambda)s(\lambda-t)d\lambda \tag{9}$$

If the input to the matched filter is $y_i(t) = s(t) + n(t)$, where n(t) represents noise, the output will be

$$y_o(t)=\int_{-\infty}^{\infty} y(\lambda)s(t_1-t-\lambda)d\lambda=R(t-t_1) \tag{10}$$

The equation above shows that the matched filter forms the cross-correlation between the received signal and a replica of the transmitted signal.

The value of the correlation integral will be greatest when the two signals being correlated are the same. By finding the value of t_1 where the greatest value of the correlation exists, the

signal that was reflected by the target can be differentiated from the noise. This procedure enables the range and velocity information to be extracted.

The Wavelet Transform

The continuous time wavelet transform (CWT) [4] produces a time-scale representation of a time function x(t) and is defined by

$$CWT_x(\tau,a)=\frac{1}{\sqrt{|a|}}\int_{-\infty}^{\infty}x(t)h^*(\frac{t-\tau}{a})dt \qquad (11)$$

The basis function h(t) is called a wavelet. These functions are obtained by scaling and shifting a prototype wavelet. The prototype wavelet h(t) is a real or complex bandpass function.
The CWT can be written as an inner product of the form

$$CWT_x=\int_{-\infty}^{+\infty}x(t)h^*_{a,\tau}(t)dt \qquad (12)$$

In this case, the above integral measures the similarity between the signal and the wavelets given by

$$h_{a,\tau}(t)=\frac{1}{\sqrt{a}}h(\frac{t-\tau}{a}) \qquad (13)$$

As previously stated, a matched filter makes use of a correlation integral in order to determine the location of velocity and range information. This integral is an inner product similar to that for the CWT. Therefore, the wavelet transform can be used to perform the correlation necessary for the detection of information in a radar system.

The detection procedure involves emitting a known signal h(t) which is then received with

a delay," τ", and a distortion or scaling," a". The delay gives information about the target's distance and information about the velocity (Doppler shift) is obtained from " a". For a wide-band signal, where the Doppler shift is not confined to a single frequency ,the CWT is given by

$$CWT_x(\tau,a)=\frac{1}{\sqrt{|a|}}\int_{-\infty}^{\infty}x(t)h(\frac{t-\tau}{a})dt \qquad (14)$$

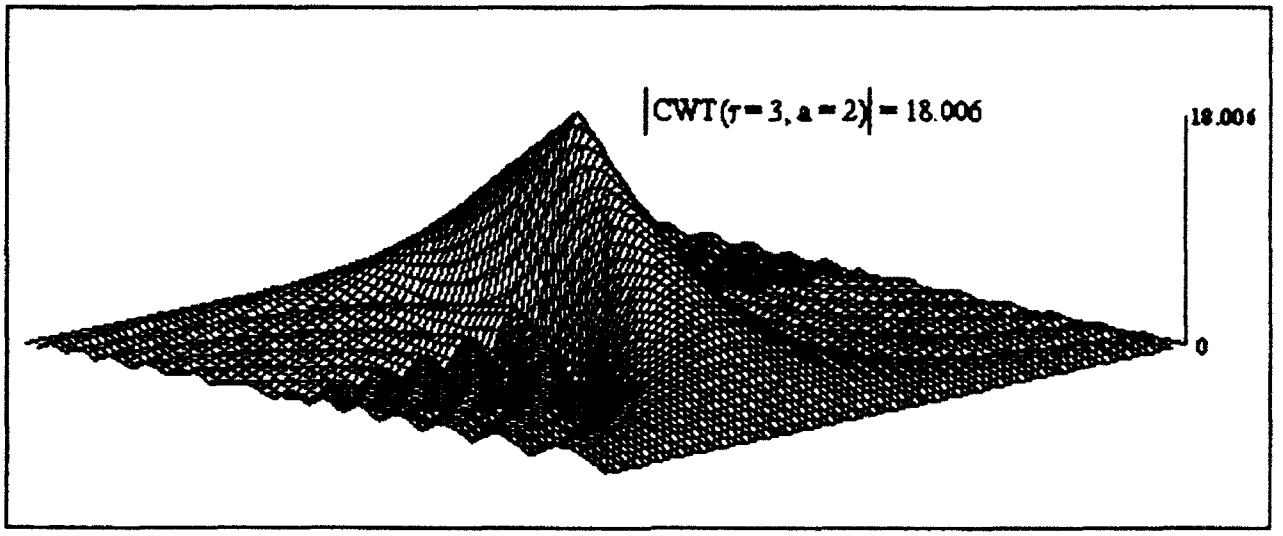
where x(t) represents the received signal. The CWT functions as a maximum likelihood estimator and the wavelet which best fits the signal is used to estimate the parameters " τ" and " a".

Example

To illustrate the ideas presented in this report, a received signal with "τ"=3 and "a"=2 was probed by a filter bank. It was desired to show that the CWT would reach its maximum at those values of "τ" and "a". Rather than use the time domain version of the CWT as given by equation (11), the frequency domain version of the CWT was used. This equation is given by

$$CWT_G(\tau,a)=\frac{1}{2\pi\sqrt{|a|}}\int_{-\infty}^{+\infty}G(\omega)H^*(\omega a)\exp(\frac{j\omega\tau}{a})d\omega \qquad (15)$$

The received signal, G(ω), is represented by a rectangle function centered at ω = 200 with a bandwidth of 40. The frequency response of the filter bank, H(ω), is given by a rectangle function centered at ω = 100 with a bandwidth of 20. Upon analysis of equation (15), the maximum value of the CWT was determined to be 18.006, located at "τ"=3 and "a"=2. This agrees with the value predicted for "τ"=3 and "a"=2.



Conclusions

Throughout this report, the wideband characteristic of phased array systems employing a continuously variable time delay has been stressed. By enabling these systems to operate in this manner, they are no longer restricted to narrowband use since the beamforming error known as squint may be eliminated. Prior to the development of the acousto-optic and fiber optic based optical processors discussed here, other methods for generating variable time delay had been lossy and impractical.

The phased array systems discussed make use of the signal processing technique of transversal filtering. The strength of transversal filters lies in their ability to be reconfigurable thereby enabling them to function in an environment that is not completely characterized or is changing. Typically, the adaptability of a transversal filter lies in the optimization of the weight vector. However, the availability of continuously variable time delay introduces another set of parameters which may be optimized to improve system performance. Future research should include determining the influence of these delay values and the manner in which they may be adjusted for decreased error.

It has been shown that, for wideband systems, a wavelet transform can perform the cross-correlation operation of a matched filter in a receive system. Due to the increased importance of

wavelet theory in the area of signal processing,it would be advantageous to incorporate wavelet methods in future transversal filter architectures for wideband phased array systems.

References

[1] H.Zmuda and E.N.Toughlian, "Adaptive Microwave Signal Processing : A Photonic Solution", Microwave Journal, Vol.35, No.2, February 1992,pp 58-71.

[2] H. Zmuda and E.N.Toughlian, " Variable Photonic Delay Line for Phased Array Antennas and RF/Microwave Signal Processing",Final Technical Report, RL-TR-91-120,Rome Laboratory, June 1991.

[3] E.N.Toughlian and H. Zmuda, "A Variable Time Delay System for Broadband Phased Array and Other Transversal Filtering Applications", Optical Engineering, to appear.

[4] O.Rioul and M.Vetterli, "Wavelets and Signal Processing", IEEE Signal Processing Magazine, Vol.8, No.4, October 1991, pp 14-38.

[5] M. Skolnik, " Introduction to Radar Systems", McGraw-Hill,Inc.,1962,1980.

[6] B. Widrow and S. Stearns, "Adaptive Signal Processing ", Prentice-Hall Inc., 1985.

[7] A. Oppenheim, A. Willsky and I. Young, "Signals and Systems" Prentice-Hall, Inc.,1983.

[8] D. Morgan, "Surface-Wave Devices for Signal Processing", Elsevier Science Publishers B.V., 1985.

[9] R.F. Mitchell, "Basics of SAW Frequency Filter Design: A Review", Computer-Aided Design of Surface Acoustic Wave Devices, Elsevier Scientific Publishing Company, 1976.

THIS PAGE INTENTIONALLY LEFT BLANK

IMPLEMENTATION OF THE ITT MULTIPLE PARAMETER SPEAKER RECOGNITION ALGORITHM ON THE SUN SPARC

Robert L. Gorsegner
Graduate Student
Department of Electrical Engineering

University of Alabama
Tuscaloosa, AL 35486

Final Report for:
Summer Research Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D.C.

September 1992

IMPLEMENTATION OF THE ITT MULTIPLE PARAMETER SPEAKER RECOGNITION ALGORITHM ON THE SUN SPARC

Robert L. Gorsegner
Graduate Student
Department of Electrical Engineering
University of Alabama

Abstract

I intend to research how speaker recognition systems work by implementing the ITT multiple parameter speaker recognition algorithm in 'C' on the Sun Sparc workstation. I will test the accuracy by using the KING database. I also intend to change the algorithm in different ways to see how accuracy is effected.

IMPLEMENTATION OF THE ITT MULTIPLE PARAMETER SPEAKER RECOGNITION ALGORITHM ON THE SUN SPARC

Robert L. Gorsegner

1 Introduction

Speaker recognition has many possible practical uses. Some of these include:

- Having computers follow who is speaking in a conversation,
- To verify users of telephone banking,
- For use in speech recognition (The ability of a computer to decode what the speaker is saying) by applying different models to different speakers the computer will be better able to identify words,
- and many others.

In general speaker recognition works in the following way. Speaker recognition algorithms convert the time domain signal into the frequency domain. Then some type of average of the frequency domain is determined for each speaker the algorithm is trained on. A speaker is then tested for every speaker in the trained set an some kind of distance between the two frequency domains is calculated. The closest distance should give the correct speaker.

2 Algorithm Description

The multiple parameter algorithm works in the following way. A frame of data is read in. The data is the filtered by a 16 order butterworth band pass IIR filter. This filter has cut off frequencies of 350Hz and 2800Hz. This filter is used to limit the frequency coming into the speaker recognition system. I have found that this filter increases the accuracy by ten percent for ten speakers and ten seconds. A speech/nonspeech detection algorithm is used next to determine if the frame is speech or just noise. In my algorithm I used the energy per frame as a silence speech detection. The

data is then windowed by a Hamming window as used by ITT [4]. Then the LPC and reflection coefficients are calculated. I found that ten coefficients is a good number to use, represented by the constant (M). ITT also used this number of Coefficients [4]. Linear Predictive Coding Coefficients are calculated by Levinson or Durbin recursion. The LPC coefficients (shown by the variable A) predict the next coefficients of data by the formula below. The variable n is the current sample data number.

$$Y_{n+1} = Y_n A_0 + Y_{n-1} A_1 + Y_{n-2} A_2 + ... + Y_0 A_{n-M} \tag{1}$$

The variable (Y) denotes the value of each point in a frame. The predictor error (pe) is the difference, totaled between the actual value of Y_{n+1} and the calculated value of Y_{n+1}. It is also the auto correlation function of the frame. The LPC cepstrum (C) is calculated from the LPC coefficients by the following.

$$C_0 = log_{10}(pe) \tag{2}$$

Let m go from 1 to M.

$$C_m = -A_m - \frac{(m-1)C_{m-1}A_1 + (m-2)C_{m-2}A2 + ... + C_1 A_{m-1}}{m} \tag{3}$$

The cepstrum represents the spectrum of the spectrum. The LPC cepstrum is used instead of the FFT cepstrum since it gives a smoother frequency response and requires less computation. The FFT cepstrum is

$$FFTcepstrum = |FFT(log_{10}(|FFT(frame)|))| \tag{4}$$

After the LPC cepstral coefficients are calculated for the frame, they are summed to a running total of all the frames. Each cepstral coefficient in a frame is multiplied by every other cepstral coefficient in that frame. These values are also kept in a running total producing an M by M matrix. By using this method it is not necessary to keep track of the cepstral coefficients for each frame. After all the frames of speech have passed through, the sums are divided by the number of frames producing an average cepstral vector ($\mathbf{X}$). The M by M matrix also is divided by the number of frames. The matrix is then subtracted by each term of the average cepstral vector multiplied by every other. This produces the covariance matrix($\mathbf{R}$) which is shown in equations 5 and 6. The variable i represent the frame number and j, k represent coefficient numbers. N is equal to the total number of frames.

$$X_j = \frac{1}{N} \sum_{i=1}^{N} C_{i,j} \tag{5}$$

$$R_{j,k} = (\frac{1}{N} \sum_{i=1}^{N} C_{i,j} C_{i,k}) - X_j X_k \tag{6}$$

The covariance matrix and average cepstral vector are calculated and saved for each speaker. To recognize an unknown speaker, the average vector must be compared against the trained speaker's average vector and covariance matrix. This is done by computing the Mahalanobis distance.

$$D = (\mathbf{X} - \mathbf{M})^T \mathbf{R}^{-1} (\mathbf{X} - \mathbf{M}) \tag{7}$$

where,

- D is the Mahalanobis distance,
- $\mathbf{X}$ is the average input vector,
- $\mathbf{M}$ is the average training vector for the speaker model,
- $\mathbf{R}$ is the covariance matrix for speaker model.

The lowest distance should be the correct speaker.

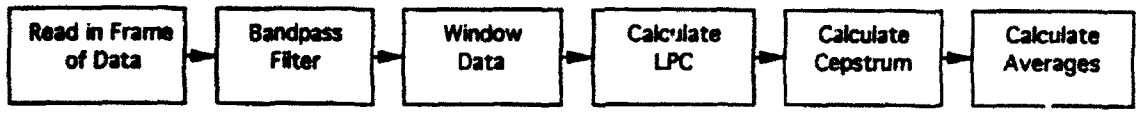


Figure 1: Flow of Algorithm

3 Speech Silence Detection Algorithm

I also included a subroutine in my program using the speech/nonspeech detection method described in "Monthly Status Report for the Project: A Modulated Model for Speaker Identification" [2]. The Speech silence algorithm goes as follows. The first five frames are considered to be silence so that a statistical analysis on the background noise can be performed. The following variables are calculated:

$$\begin{array}{lll} \text{(Energy Threshold)} & T_1 = ave + \alpha_1 ste & \alpha_1 > 0 \\ \text{(Zero Crossing Rate)} & T_2 = avz + \alpha_2 stz & \alpha_2 > 0 \\ \text{(Magnitude Threshold)} & T_3 = avm + \alpha_3 stm & \alpha_3 > 0 \end{array}$$

where,

- ave : average energy per frame
- avz : average number of zero crossings per frame
- avm : average magnitude per frame
- ste : standard deviation of energy
- stz : standard deviation of zero crossing rate
- stm : standard deviation of magnitude

According to the designers of the algorithm a good choice of α_1, α_2, and α_3 is three which is the value I used in my program.

After the speech/silence statistics are calculated the rest of the program data is tested.

If ($E >= T_1$) or ($Z >= T_2$) or ($M >= T_3$) then it is classified as speech. Where,

- E : Energy of the frame
- Z : Number of zero crossings
- M : Average magnitude of the frame

I did not use this method since in the KING database[1] the first few frames sometimes were abnormally silent and other times had voice in them. This caused the algorithm to not work. Instead I preset a value for $T_1 = 2500(framesize)$. I did not use T_2, or T_3 to test for speech. The code that calculates the values of T_1, T_2 and T_3 is commented out.

4 Program Usage

To train the speaker identification program for a specific speaker use the format in figure 2. This creates or appends to a parameter file. The name of the parameter file has the window length and type appended to it. The types of windows available are (0) for rectangular,(1) Gaussian, (2) Hamming, and (3) for raised cosine. The program will train on the utterance until either the specified number of frames has been reach or the end of the file. If it hit the end of file prematurely no warning will be given.

To test an utterance the user changes the 'Train/Test' flag to 1. The program will show the distance between each speaker and the winning distance.

5 Program Description

The Speaker Recognition system has a training mode and a testing mode. In the training mode LPC cepstral coefficients are generated for the frames of speech data and stored in a parameter

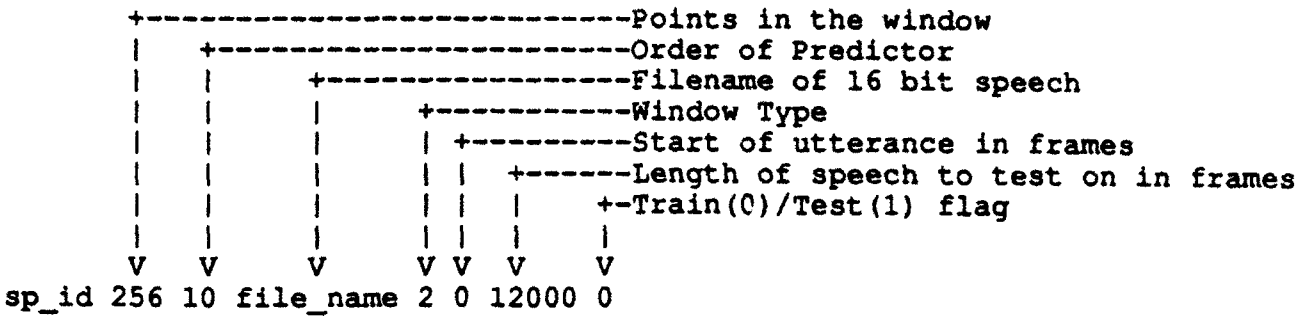


Figure 2: Shows format to call speaker identification program.

file. The testing mode generates coefficients for the unknown speaker and compares them to the coefficients in the parameter file. It then reports the results to the user.

The ITT speaker identification system is coded in 'C' shown by the flowchart in figure 3 [4]. The first block allocates memory for the covariance matrix, average vector, and window values. Next the speaker identification program gets the arguments from the command line. The arguments include the test speaker file, window or frame size, window type, order of predictor, starting point, and length of speech data to train on. The window types that are available in the program are rectangular, Gaussian, Hamming, and raised cosine. For accuracy the same window must be used for training and testing. The window is then calculated and stored in memory. This is done so that the window does not have to be calculated for each frame.

The Calculate Cepstral Covariance Matrix and Average vector block of code (figure 4)reads a 16-bit speech data file and calculates the average and the covariance matrix of the cepstral coefficients. The first step is to read a frame of speech. The frame is then filtered by using a subband filter and windowed. If the energy of the frame is not above a certain value a new frame is read in. The linear predictive coding(LPC) coefficients and the reflection coefficients are calculated by using the subroutine found in "Voice and Speech Processing" by T. Parsons. Some of the variables in this

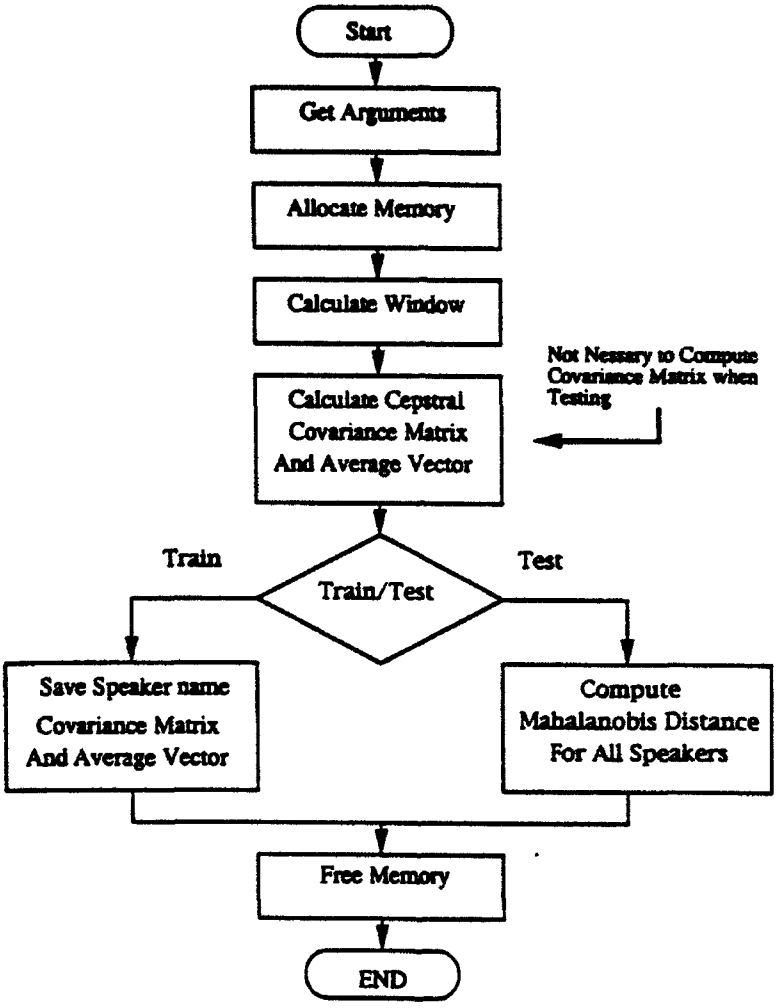


Figure 3: Main flow of the program.

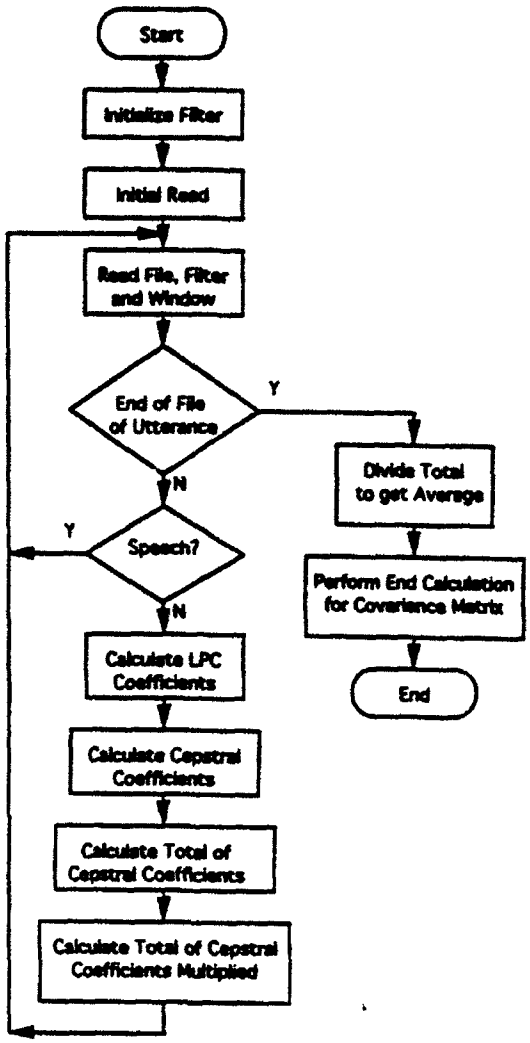


Figure 4: Calculation of average vector and covariance matrix.

subroutine are required to be double precision floating point numbers (8 bytes) instead of the single precision floats. The LPC coefficients are then passed to a subroutine found in "Linear Prediction of Speech"[3] to calculate the cepstral coefficients. The zeroth cepstral coefficient is calculated but not used. The cepstral coefficients are totaled. Each cepstral coefficient is multiplied by every other cepstral coefficients for a certain frame and a total is kept in the covariance matrix. This process is done until the end of the speech data file or the set number of frames has been processed. At the end, the cepstral coefficients are divided by the number of frames to produce an average of the cepstral coefficients. The covariance matrix is also divided by the number of elements and then subtracted by the product of the average cepstral coefficients. See the Covariance formula for better explanation.

When training the speaker identification system writes or appends if the parameter file already exists the speaker name, average vector and covariance matrix in ASCII. This can be viewed by use of the 'more' or 'pg' UNIX commands.

During testing the parameters for each trained speaker are read. The average vector of the unknown and each trained speaker are subtracted. This value is multiplied by the inverse of each trained speakers' covariance matrix. Then it is multiplied by the transverse of the difference of the average vectors. This calculation results in a scalar value called the Mahalanobis distance. This process is done for all trained speakers. The lowest of these distances should match the correct speaker.

6 Test Results

I used the KING database for testing. The KING database has fifty speakers. Each of these speakers spoke in ten sessions which are about forty seconds each.[1]. Each session is recorded wide

Known	Identified Speaker									
Speaker	1	2	3	4	5	6	7	8	9	10
1	3	0	0	1	0	0	0	0	1	0
2	3	6	0	0	0	0	0	0	0	2
3	0	0	6	0	0	0	0	0	0	1
4	0	0	0	2	0	0	0	0	0	0
5	0	0	0	0	6	1	0	0	0	0
6	0	0	0	1	0	3	0	0	2	0
7	0	0	0	1	0	0	6	0	0	1
8	0	0	0	0	0	2	0	6	0	0
9	0	0	0	1	0	0	0	0	3	2
10	0	0	0	0	0	0	0	0	0	0

Table 1: Confusion matrix showing performance of speaker identification algorithm using ten speakers and ten second utterances.

band through a narrow band channel. It was recorded at both ends of a telephone line. In my tests I have only used the wide band speech. In my first test I used sessions one through three of the wide band to train the speaker identification system. I tested on six ten-seconds intervals of sessions four and five. I used ten cepstral coefficients and a 16 order butterworth bandpass filter with cutoff frequencies of 350Hz and 2800Hz. I generated a confusion matrix (Table 1) to show the results. To generate the confusion matrix I modified my 'C' program to do batch runs. The accuracy over these runs was 68 percent. I also kept track of the average distance (Table 2) for each utterance. Testing on thirty two different two second intervals is shown in Table 3. An accuracy of 53 percent was obtained.

An error that I had in my program which is now fixed showed something interesting. I accidently used a window that was normal for one half of the frame and inverted for the other half. I found only a 18 percent reduction in correct identification for this weird window. The results for using ten speakers, and ten seconds are found in table 4. The accuracy of this was 50 percent.

Known Speaker	Identified Speaker 1	2	3	4	5	6	7	8	9	10
1	3.838	5.818	1.078	1.020	1.672	1.174	1.013	1.221	0.624	5.465
2	3.948	0.307	3.279	2.860	2.887	2.803	4.983	5.067	2.844	4.762
3	3.438	4.630	0.435	1.427	1.042	1.094	1.807	1.326	1.072	4.701
4	7.063	5.476	1.501	0.846	3.723	1.994	0.756	1.480	1.231	7.918
5	4.032	3.236	1.588	2.285	0.278	0.716	3.767	3.090	1.228	5.454
6	3.583	3.200	1.061	1.523	0.962	0.426	1.939	2.281	0.447	4.234
7	5.727	5.970	1.909	1.197	5.262	3.451	0.299	1.582	3.448	5.038
8	4.385	5.585	0.981	1.099	1.647	0.817	2.154	0.700	0.829	5.657
9	4.134	3.874	1.055	0.952	1.236	0.695	1.518	2.112	0.376	5.238
10	4.826	1.512	1.404	1.543	1.661	0.889	1.874	3.030	0.601	5.324

Table 2: Confusion matrix showing average Mahalanobis distances using ten speakers and ten second utterances.

Known Known Speaker	Identified Speaker Identified Speaker 1	2	3	4	5	6	7	8	9	10
1	16	0	2	5	0	0	2	1	2	1
2	11	27	0	0	0	0	0	0	0	9
3	2	0	18	0	7	2	0	6	0	4
4	0	0	3	10	0	0	3	4	0	2
5	0	2	0	0	25	9	0	0	3	0
6	3	0	2	2	0	14	0	0	13	2
7	0	0	0	6	0	0	27	2	0	5
8	0	0	6	2	0	5	0	19	0	4
9	0	0	0	6	0	2	0	0	12	2
10	0	3	1	1	0	0	0	0	2	3

Table 3: Confusion matrix showing performance of speaker identification algorithm using ten speakers and two second utterances.

Known	Identified Speaker									
Speaker	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	6	0	0	0	0	0	0	0	0
3	0	0	6	0	0	0	0	3	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	6	3	0	0	0	0
6	0	0	0	0	0	3	0	0	0	0
7	0	0	0	0	0	0	3	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	6	0	0	6	0	0	3	3	6	6
10	0	0	0	0	0	0	0	0	0	0

Table 4: Confusion matrix showing performance of speaker identification algorithm using ten speakers and ten second utterances using weird window.

7 Future Improvements

I intend to continue my work on speaker recognition as part of my masters thesis at the University of Alabama. I intend to try several things to improve performance. These include using reflection coefficients, using FFT-cepstral coefficients, varying the order of predictor, trying different front end filters and using a clustering method such a K-means. I have written a program using K-means codebooks but I have not debugged it yet. I also intend to look for possible ways of speeding up the code, for instance changing floating point numbers to integers, or implementation on a dedicated DSP board. I also made a X-window interface for the program. Currently I do not have all the options of the windowed version working. It is designed using a interface design tool called Guide [5]. This program shows the following window with all the options as shown. The window interface version is very user friendly.

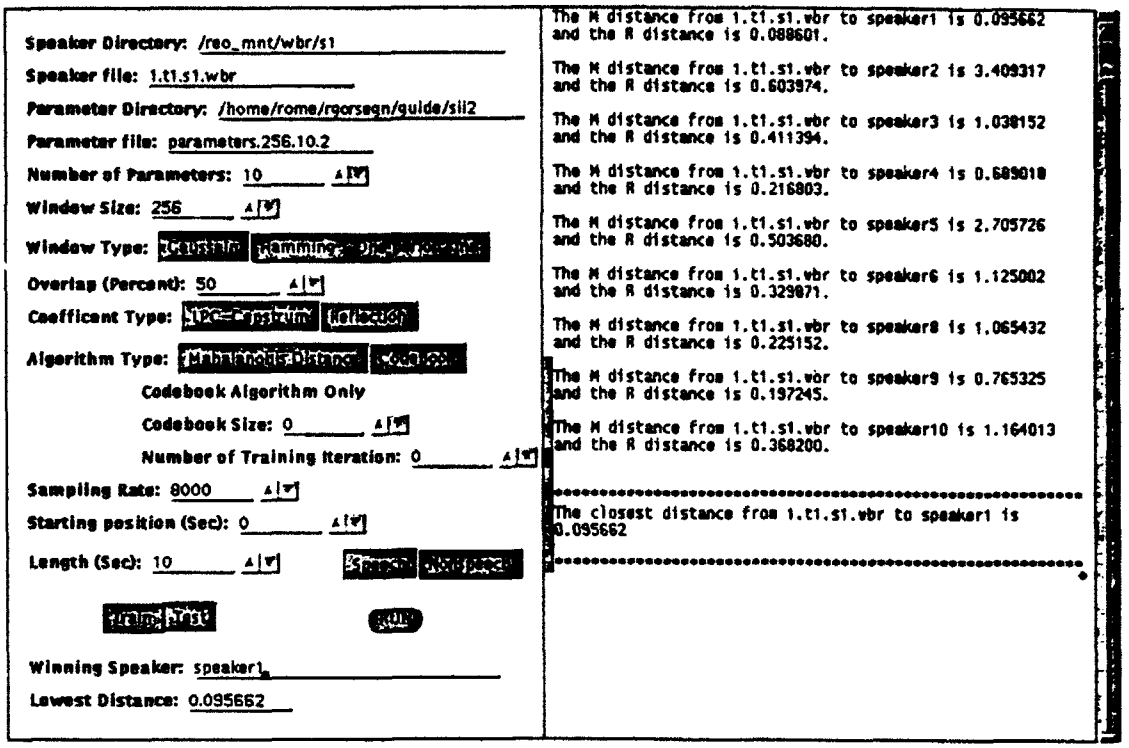


Figure 5: Picture of window interface.

References

[1] A. Higgins, E. Wrench, L. Bahler, J. Porter, D. Schmoldt, M. Lipps (November 1991) "*Speaker Identification and Recognition*" Final Report Contract 88-F744200-000, ITT Aerospace/Communications Division, San Diego,CA.

[2] Richard J. Mammone (March 13,1992) "*Monthly Status Report for the project: A modulation Model for Speaker Identification*" Rome Laboratory, NY.

[3] Markel, Gray (1976) "*Linear Prediction of Speech*" Springer-Verlag, NY.

[4] J. Naylor, E. Wrench and R. Wolhford (September 1986) "*Multi-Channel Speaker Recognition*" RADC-TR-85-280 Final Technical Report, Rome Air Development Center, NY.

[5] "*OpenWindows Developer's Guide 1.1: Users Manual*" (June 1990) Sun Microsystems, USA.

[6] T. Parsons (1987) "*Voice and Speech Processing*" McGraw Hill, New York.

MATHEMATICAL DESCRIPTION, COMPUTER SIMULATION AND ANALYSIS OF A POINTING, ACQUISITION AND TRACKING SYSTEM FOR OPTICAL INTERSATELLITE CROSSLINKS

Carl R. Herman
Graduate Student
Department of Electrical Engineering

Binghamton University
P.O. Box 6002
Binghamton, New York 13902-6002

Final Report for:
Graduate Student Research Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Griffiss Air Force Base, New York

September 92

MATHEMATICAL DESCRIPTION, COMPUTER SIMULATION AND ANALYSIS OF A POINTING, ACQUISITION AND TRACKING SYSTEM FOR OPTICAL INTERSATELLITE CROSSLINKS

Carl R. Herman
Graduate Student
Department of Electrical Engineering
Binghamton University

Abstract

The mathematical model of a pointing, acquisition and tracking (PAT) optical intersatellite crosslink system is developed. It describes a laboratory prototype, created to demonstrate various aspects of rapid-retargeting bi-directional laser communications between independent space-based stations. The model, obtained by the detailed analysis of system hardware, represents the dynamic properties of the major optical, electrical and electro-mechanical system components. A computer simulation program, utilizing the mathematical model, is developed. The validity of the model is assured by comparing the experimental and simulated system responses to various operational conditions. The developed model provides a versatile and cost-effective alternative for verification of novel concepts in optical intersatellite crosslinking, and specifically, advanced control strategies.

MATHEMATICAL DESCRIPTION, COMPUTER SIMULATION AND ANALYSIS OF A POINTING, ACQUISITION AND TRACKING SYSTEM FOR OPTICAL INTERSATELLITE CROSSLINKS

Carl R. Herman

INTRODUCTION

The PAT system, created by Ball Aerospace Corporation, was created to demonstrate three concepts in rapid-retargeting, bi-directional laser communications between independent space-based stations,

1) an open-loop pointing system capable of rapid selection and sequential communication to a large number of optical stations,

2) an open-loop pointing system with multiple beam directors and beam switching capabilities allowing multiplexing in time and beam slewing,

3) a closed-loop pointing system with beam switching that allows for sequential establishment of a large number of two-way, closed-loop communication links.

The system consists of one stationary terminal, one mobile remote terminal, and one stationary remote terminal, representing three satellites in low earth orbit, medium earth orbic, or geosynchronous earth orbit. The primary station includes its own electronics rack and computer, while the remote stations share a common electronics rack and computer.

In addition to the demonstrations, operation of the optical system for each terminal is characterized by a set of pretests that measure critical performance parameters (Table 1).

Acquisition beam power
Acquisition beam irradiance profile
Communication beam power
Communication beam irradiance profile
Bit Error Rate (BER) vs. received power
Average BER at primary station
Average BER at remote stations
Average BER while tracking a moving target

Table 1. Pretests performed before operation of the PAT system.

DESCRIPTION OF THE SYSTEM HARDWARE

The primary station hardware includes a Hewlett Packard Model 330 computer, an electronics rack housing the support electronics for the electro-optic and electro-mechanical components, and an optical table where the optical components are mounted.

The primary station optic and electro-optic systems can be categorized into six groups: lasers, detectors, beam conditioners, ray optics, filters, and beam directors (see Figure 1). The two lasers of the primary station are semiconductor lasers with optical output powers of 10 mW and 100 mW, and center wavelengths of 830 nm and 800 nm respectively. There are three types of optical detectors in the primary station, two types for tracking purposes and one for communications. The tracking detector is a quadrant silicon avalanche photodiode, the acquisition detectors are improved tetra-lateral position sensitive devices (PSD) and the communications detector is a silicon avalanche photodiode. The beam conditioners and ray optics assure the required characteristics of the communication and beacon laser beams for detection and transmission. The optical filters are used to

separate these beams so that each can be processed individually. The beam directors (CSM 1 and CSM 2) and the fast steering mirror (FSM) are used to direct the transmitted beams and to position the received beams.

The primary station electric and electro-mechanic systems can be categorized into five groups: the computer, the computer interface, the fine steering subsystem, the coarse steering subsystem, and the wide field of view (WFOV) subsystem. The computer interface consists of a HP general purpose input/output (GPIO) board and interface electronics. Both the fast steering subsystem and the coarse steering subsystem are analog closed-loop control circuits activated by the computer. The WFOV camera is stationary and therefore contains no control system. The camera's sole purpose is to detect the remote stations beacon light emitting diodes (LEDs) and provide initial pointing information which is directly input to the computer. The function of each subsystem and the interaction between subsystems is controlled by the computer (Appendix A).

The remote stations consist of a HP Model 330 computer, an electronics rack housing the support electronics for electro-optic and electro-mechanic components, and two optical tables, one stationary and one mobile, where optical components are mounted.

Like the primary station, the remote stations optic and electro-optic systems are composed of lasers, detectors, beam conditioners, ray optics, filters, and beam directors (see Figure 2). The two lasers in the remote station are semiconductor lasers

with optical output powers of 10 mW and 50 mW, and center wavelengths of 780 nm and 810 nm respectively. The communications detector is a silicon avalanche photodiode and the acquisition detector is a silicon photodiode. Each station contains beam conditioners and filters to prepare the communication and beacon laser beams for detection and transmission. As with the primary station, the optical filters isolate the four wavelengths of light in the system so that each can be processed individually.

As with the primary station, the function of each subsystem and the interaction between subsystems is controlled by the computer (Appendix A).

PRINCIPLE OF SYSTEM OPERATION

The control system in the PAT station has two principal functions: to point the communications laser from the primary station to remote stations, and to position the communications laser from the remote stations on the communication receiver of the primary station. The operation of this control system is based on three stages of positioning; initial coarse steering mirror (CSM) orientation, coarse steering, and fine steering.

The first stage uses the "very coarse" information on target position obtained from the WFOV camera to point the primary station lasers at the remote stations. The image created by the beacon LEDs located on each of the remote stations is captured by the optics of the WFOV camera. This image is analyzed and the coordinates of the targets are input directly into the computer. The computer uses this information to calculate the displacements

of the coarse steering mirrors required for the initial CSM orientation. At this stage the primary station beams are pointed in the direction of the remote stations and the communication and beacon beams coming from the remote stations will be used for fine positioning of the incoming communication beams. This initial orientation of the CSMs does not rely on feedback information on beam position and is therefore open-loop in nature.

The next stage implies positioning the communication laser based on the signals generated by the acquisition detectors which provide information on the deviation of the laser position from the center of the PSD. These signals are fed back directly to the servo-control electronics of the CSMs. In this fashion the closed-loop control system adjusts the orientation of the CSMs to maintain the communication laser position in the center of the PSDs. When a communication laser is positioned in the center of a PSD it is then said to be "acquired".

The final stage in positioning the laser uses the incoming beacon laser from the remote stations to provide precise fine position information. The beacon laser is narrowed by a 40X telescope (see Figure 1) and then focused onto a quadrant silicon avalanche photodiode. The positioning error information, obtained by the quadrant detector, is fed back directly to the servo-control electronics of the FSM. In this fashion the closed-loop control system continually adjusts the orientation of the FSM to maintain the beacon laser position in the center of the quadrant detector.

The design of the optical system assures that if the beacon

laser is positioned correctly on the quadrant detector the communications laser will be positioned on the communications receiver. Thus the communications laser is used for coarse positioning and the beacon laser is used for fine positioning. When a target is moving, the quadrant detector provides the information on position change and adjusts the FSM to compensate.

ENVIRONMENTAL EFFECTS

Optical interference from external sources is a major concern in the operation of space-based laser communication systems. To simulate optical noise in the PAT system "white" light is introduced into the communication laser's path by a fiber optic link (see Figure 1). White light simulates typical radiation spectra encountered in space-based systems.

Platform vibration (jitter) in satellite communication systems is one of the most important issues. Internal cyclic moments created by gyroscopes, servo-mechanisms, etc., thermal expansion/contraction of the platform body, and gravitational field effects create a constant source of platform vibration. Other sources of vibration exist that are not cyclic in nature and create dynamic, unpredictable perturbations in the stability of the platform. Table 2 presents a list of possible sources of platform vibration or "jitter". The dominance of any one source of vibration will depend entirely on the physical nature of the satellite and on the specifics of operation. Simulated platform jitter is introduced into the PAT system by a galvanometer placed

Sources of on-board mechanical vibration
Gyroscopes, servos, etc.
Earth's central gravitational field
Elastic forces of tension and bending
Effects of initial conditions
Ellipticity of orbit
Earth oblateness effects
Solar and lunar gravity
Solar radiation pressure
Micrometeoroid impacts
Thermal bending

Table 2. Possible sources of platform vibration on a satellite.

in the optical path of the primary station (see Figure 1). The spectrum of vibration frequencies introduced by the galvanometer is dominated by low frequencies with a cutoff of about 100 Hz.

MATHEMATICAL DESCRIPTION OF THE PAT SYSTEM

Functional and block diagrams are used as graphical aids to demonstrate hardware configuration and information flows. Explicit mathematical formulas for each system block are derived from circuit analysis (Appendix C), ray optics, electro-optic material analysis and manufacturer specifications depending on the nature of the block.

The hardware configuration and flow of information for the overall system is shown in the functional diagram in Figure 3. It features three major feedback loops, one in the fine steering subsystem, one in the coarse steering subsystem, and one formed by the WFOV subsystem and the computer interface.

Figure 4 shows the fine steering functional diagram which actually contains two "local" information feedback loops, one for optical feedback and one for mirror position feedback. Note that

although both feedback loops are directly dependant upon the FSM angular position, they do not use the same control law. The error signal for the optical loop is derived from the laser's position on the quadrant detector, while the error for the position loop is obtained from the mirrors' angular position.

The coarse steering functional diagram is shown in Figure 5 and shows four "local" information feedback loops. The four loops bring information from the galvanometers, the fine steering differential impedance transducers (DITs) interface, the optical PSDs and the WFOV camera via the D/A converter in the servo -electronics block. Each of the four loops brings information back to the servo-electronics block for processing. The CSM assembly contains its own electronics and internal feedback system which is not shown in this diagram. For more information on the CSM block see Appendix B.

The WFOV functional diagram is shown in the Figure 3. The optical information obtained by the WFOV camera is generated by an image of the beacon LEDs on the remote stations and not by laser position as is the case for the acquisition and tracking detectors. This information is directly introduced to the PC via the GPIO interface and is used for adjusting the CSM position either directly or by adding position information to the other three feedback signals. In either mode of operation the WFOV functional diagram does not contain "local" feedback loops.

The fine steering block diagram is shown in Figure 6 and operates in two modes controlled by switches S_{F1} and S_{F2}. The

computer controls both switches using the enable track line from the GPIO board. Fine tracking is enabled by switching S_{F1} and S_{F2} to position 1 thereby using the optical feedback information to control the positioning of the fine steering mirror. When the switches are in position 2 the DITs feedback information is used to control the mirror position. The mathematical description of each block in the fine steering block diagram follows.

Fine Steering Transfer Functions

Compensator:

$$G_{F1} = -12.0763(s+60.7275)/(s+54.9665),$$

$$G_{F2} = -263600(s+3078.8)/((s+71808)(s+11302)),$$

$$G_{F3} = -2308.1/(s+2308.1).$$

Coordinate Rotation:

$$G_{F4} = -2.004 \times 10^6/(s+2.004 \times 10^6)$$

Current Driver, Reaction Mass Actuator, Steering Mirror Actuator:

$$G_{F5} = 17.0983 \times 10^6/(s^2+672.8s+17.0983 \times 10^6)$$

Note: s - is Laplace operator (symbol of differentiation)

Fine Steering Static Relationships

Primary SIM AQC Main, R_{F1}:

$$X_{FO}(t) = [(I_{QX1}+I_{QX2})-(I_{QX3}+I_{QX4})] \;/\; [I_{QX1}+I_{QX2}+I_{QX3}+I_{QX4}]$$

$$Y_{FO}(t) = [(I_{QY2}+I_{QY4})-(I_{QY1}+I_{QY3})] \;/\; [I_{QY1}+I_{QY2}+I_{QY3}+I_{QY4}]$$

Quadrant Detector, R_{F2}:

$$I_{QX1} = I_{QY1} = \begin{cases} I_o/8(R_{max}-(|X_Q-X_{RX1}|^2+|Y_Q-Y_{RX1}|^2)^{1/2}), & \text{if } X_Q<X_{RX} \text{ and } Y_Q>Y_{RX} \\ 0, \text{ Otherwise} \end{cases}$$

$$I_{QX2} = I_{QY2} = \begin{cases} I_o/8(R_{max}-(|X_Q-X_{RX2}|^2+|Y_Q-Y_{RX2}|^2)^{1/2}), & \text{if } X_Q<X_{RX} \text{ and } Y_Q<Y_{RX} \\ 0, \text{ Otherwise} \end{cases}$$

$$I_{QX3} = I_{QY3} = \begin{cases} I_o/8(R_{max}-(|X_Q-X_{RX3}|^2+|Y_Q-Y_{RX3}|^2)^{\frac{1}{2}}), & \text{if } X_Q>X_{RX} \text{ and } Y_Q>Y_{RX} \\ 0, & \text{Otherwise} \end{cases}$$

$$I_{QX4} = I_{QY4} = \begin{cases} I_o/8(R_{max}-(|X_Q-X_{RX4}|^2+|Y_Q-Y_{RX4}|^2)^{\frac{1}{2}}), & \text{if } X_Q>X_{RX} \text{ and } Y_Q<Y_{RX} \\ 0, & \text{Otherwise} \end{cases}$$

Geometry, Ray Optics, R_{F3}:

$$X_Q(t) = (d+\Delta d)[\sin(\alpha_{FXo}+2\Delta\alpha_{FX}(t)-\sin(\alpha_{FXo})]$$

$$Y_Q(t) = (d+\Delta d)[\sin(\alpha_{FYo}+2\Delta\alpha_{FY}(t)-\sin(\alpha_{FYo})]$$

Kaman DITs Electronics, R_{F4}:

$$X_{FP} = X_{FPo}(|I_{DITX1}| - |I_{DITX2}|)$$

$$Y_{FP} = Y_{FPo}(|I_{DITY1}| - |I_{DITY2}|)$$

Differential Impedance Transducers, R_{F5}:

$$I_{DITX1} = \kappa_1(L_{OX}+L_{MX}\sin(\alpha_{FX}(t)))\sin(\omega_{OSC})$$

$$I_{DITX2} = -\kappa_1(L_{OX}+L_{MX}\sin(\alpha_{FX}(t)))\sin(\omega_{OSC})$$

$$I_{DITY1} = \kappa_1(L_{OY}+L_{MY}\sin(\alpha_{FY}(t)))\sin(\omega_{OSC})$$

$$I_{DITY2} = -\kappa_1(L_{OY}+L_{MY}\sin(\alpha_{FY}(t)))\sin(\omega_{OSC})$$

The coarse steering control block diagram of the CSM subsystem is displayed in Figure 7. The computer operates switches S_{C1} through S_{C10}, controlled through the GPIO board, to change the mode of operation between pointing, coarse tracking, and fine tracking. Pointing is enabled by opening all switches and selecting S_{C5} and S_{C10} to position 2 thereby using the WFOV feedback information from the computer to control the positioning of the coarse steering mirrors. Coarse tracking is enabled when switches S_{C5} and S_{C10} are in position 1 and the LEC and Galvo feedback switches are closed. Fine tracking is enabled when switches S_{C1}, S_{C6}, S_{C2}, and S_{C7} are

closed, closing the DIT and Galvo information loops.

The mathematical representation of each block in the coarse steering block diagram follows.

Coarse Steering Transfer Functions

Pre-filters:

$$G_{C1} = -6666.7/(s+550.964)$$

$$G_{C2} = -200401/(s+10^6)$$

$$G_{C3} = -10^6/(s+10^6)$$

Coarse Steering Command Electronics:

$$G_{C4} = -10^6/(s+10^6)$$

$$G_{C5} = -0.02315(s+701.459)/(s+325.489)$$

$$G_{C6} = -60.6061/(s+60.6061)$$

CSM Control Electronics, Steering Mirror Actuator:

$$G_{C7} = 756.072 \times 10^3/(s^2+395.637s+756.072 \times 10^3)$$

Coarse Steering Static Relationships

Acquisition Main Electronics R_{C1}:

$$X_{CO}(t) = [(I_{PSD3}+I_{PSD4})-(I_{PSD1}+I_{PSD2})] \;/\; [I_{PSD1}+I_{PSD2}+I_{PSD3}+I_{PSD4}]$$

$$Y_{CO}(t) = [(I_{PSD1}+I_{PSD3})-(I_{PSD2}+I_{PSD4})] \;/\; [I_{PSD1}+I_{PSD2}+I_{PSD3}+I_{PSD4}]$$

Position Sensitive Device, R_{C2}:

$$I_{PSD1}=\tfrac{1}{2}I_o[(L-X_A)/2L + (L+Y_A)/2L]$$

$$I_{PSD2}=\tfrac{1}{2}I_o[(L-X_A)/2L + (L-Y_A)/2L]$$

$$I_{PSD3}=\tfrac{1}{2}I_o[(L+X_A)/2L + (L+Y_A)/2L]$$

$$I_{PSD4}=\tfrac{1}{2}I_o[(L+X_A)/2L + (L-Y_A)/2L]$$

Geometry, Ray Optics, R_{C3}:

$$X_{PSD}(t) = d[\sin(\alpha_{CXo}+2\Delta\alpha_{CX}(t)-\sin(\alpha_{CXo})]$$

$$Y_{PSD}(t) = d[\sin(\alpha_{CYo}+2\Delta\alpha_{CY}(t)-\sin(\alpha_{CYo})]$$

Current to Voltage Conversion, R_{C4}:

$$X_{CP} = X_{CPo}(|I_{CAPX1}| + |I_{CAPX2}|)$$

$$Y_{CP} = Y_{CPo}(|I_{CAPY1}| + |I_{CAPY2}|)$$

Capacitive Rotation Sensors, R_{C5}:

$$I_{CAPX1} = \kappa_2[I_{Xo}+I_{Xm}\sin(\alpha_{CX}(t))]$$

$$I_{CAPX2} = \kappa_3[I_{Xo}+I_{Xm}\sin(\alpha_{CX}(t))]$$

$$I_{CAPY1} = \kappa_4[I_{Yo}+I_{Ym}\sin(\alpha_{CY}(t))]$$

$$I_{CAPY2} = \kappa_5[I_{Yo}+I_{Ym}\sin(\alpha_{CY}(t))]$$

COMPUTER SIMULATION PROGRAM

The "PAT Simulation Program" which implements the mathematical model is written in about 1300 lines of Turbo Pascal computer code (Appendix D). This program simulates the dynamic responses of the coarse and fine steering subsystems, positions of the acquisition and tracking lasers on the PSD and the quadrant detector respectively, and the effects of random and periodic disturbance signals.

The PAT simulation program allows the user to customize the spectrum of the disturbance signal to simulate a large variety of satellite configurations and environments. White noise disturbance signals are contained in a user adjustable low frequency envelope and can be adjusted in magnitude to fit any typical low-pass frequency spectrum. An adjustable "frequency spike" can be placed at any frequency below 1 KHz to simulate a single frequency cyclic disturbance signal (Figure 8). After the user specifies the disturbance signal, the program computes and displays the disturbance both in the time- and frequency-domains. This provides

an opportunity to verify the disturbance signal and change it if necessary. The PAT simulation program has both one- and two-dimensional display capabilities, as shown in Figures 9 through 13. Each plotting mode features the dynamic responses of the coarse and fine steering mirrors, specifically X and Y positions of the laser beams. However, in the first model X and Y positions are displayed separately as feedback signals of the appropriate control loops, while the second mode displays laser position which represents the feedback control signals for both the acquisition and tracking detectors. The magnitude of the control signals is directly proportional to the initial X and Y coordinates of the remote stations with respect to the orientation of the primary station. The two flat lines represent the X and Y control signals' final rest position after the target has been acquired.

The two dimensional simulation shows the position of the communication and beacon lasers on the acquisition and tracking detectors, as shown in Figures 11 and 12. This simulation represents a static translation from mirror angular position to laser position on the detectors and is directly proportional to the optical feedback signals X_{CO}, Y_{CO}, X_{FO}, and Y_{FO}.

MODEL VERIFICATION

To test the accuracy of the defined mathematical model a step voltage signal was introduced into the reference input to the fine steering subsystem. Figure 14 shows an oscilloscope photograph of the obtained step response as measured from the X position DITs interface. Comparing this to the model response in Figure 13 shows

good correlation. The same test was performed on the coarse steering subsystem with similar results.

BACKGROUND LITERATURE

1. Eric A. Swanson and James K. Roberge, "Design considerations and experimental results for direct-detection spatial tracking systems", Optical Engineering, Vol. 28 No. 6. June 1989.

2. R. Dumas and B. Laurent, "System test bed for demonstration of the optical space communications feasibility", SPIE Vol. 1218, Free-Space Laser Communication Technologies II, 1990.

3. J. D. Barry and G. S. Mecherle, "Beam pointing error as a significant design parameter for satellite-born, free-space optical communication systems", Optical Engineering, Vol. 24 No. 6. December 1985.

4. "An experimental spatial acquisition and tracking system for optical intersatellite crosslinks", Technical Report prepared for the Department of the Air Force under contract F19628-90-C-0002, December 1991.

5. "The flexural behavior of PACSAT in orbit", Prepared for the Defense Advanced Research Projects Agency by Rand Corporation, February 1983.

Primary Station Optical Schematic

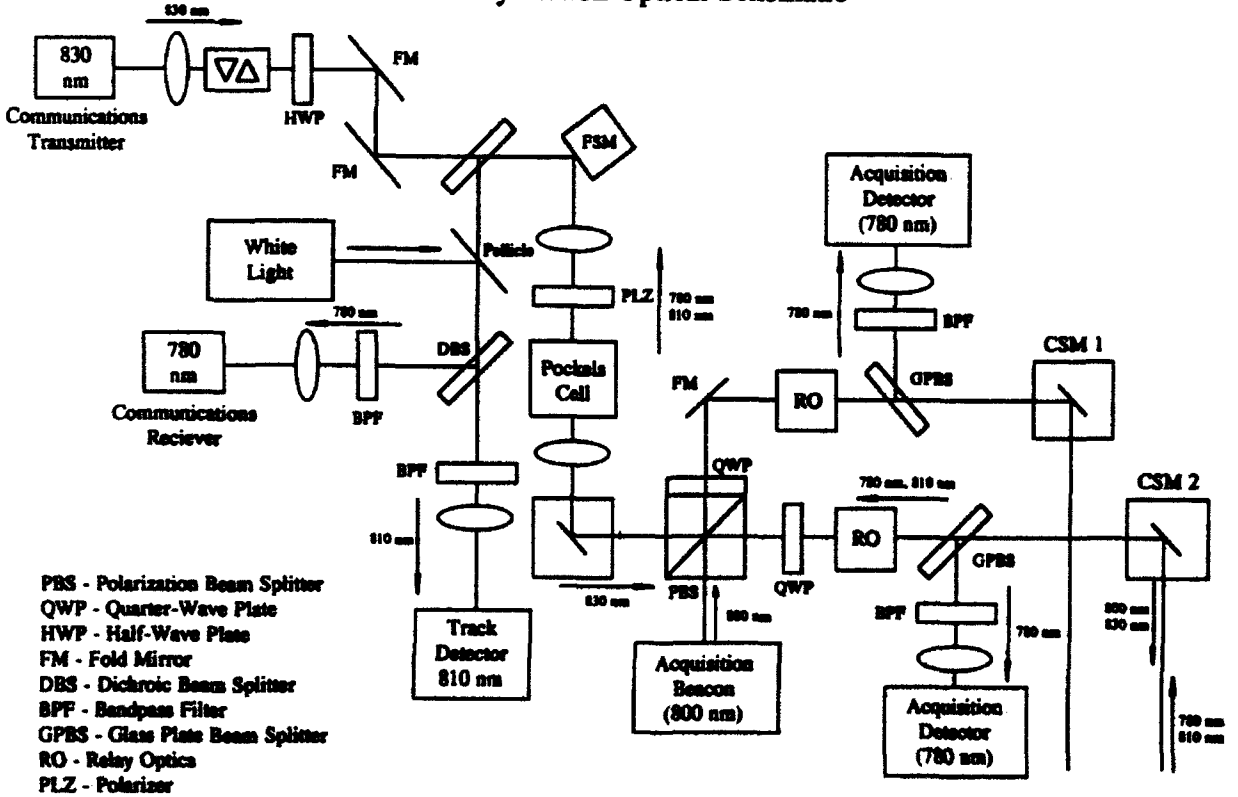


Figure 1. Primary station optical schematic.

Remote Station Optical Schematic

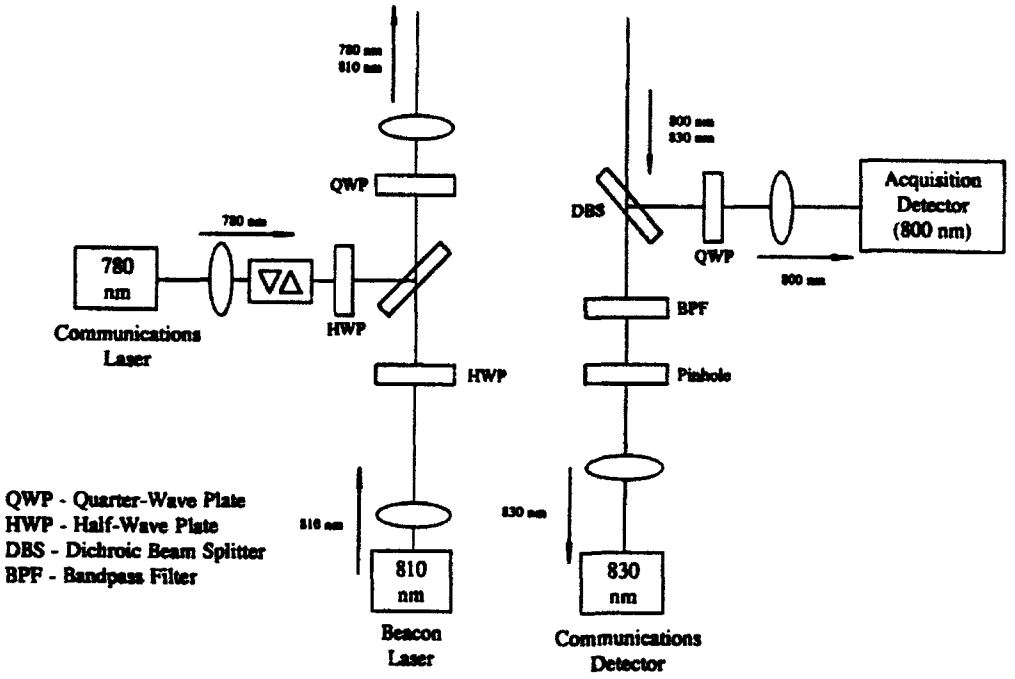


Figure 2. Remote station optical schematic.

Over-All System Functional Diagram

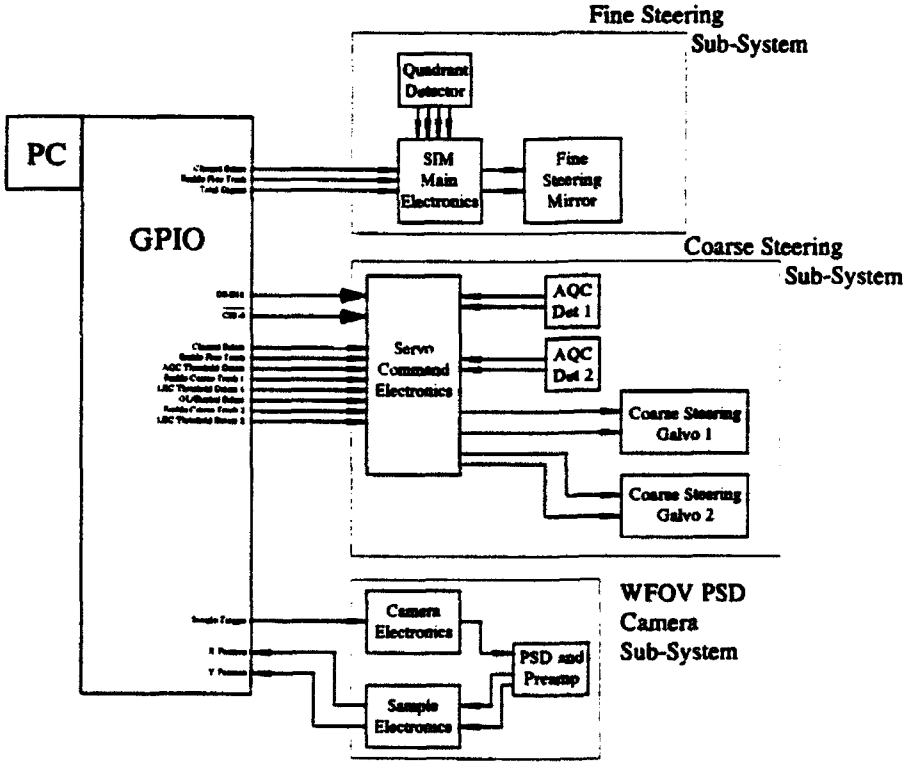


Figure 3. Overall Functional block diagram for the PAT system.

Fine Steering Functional Diagram

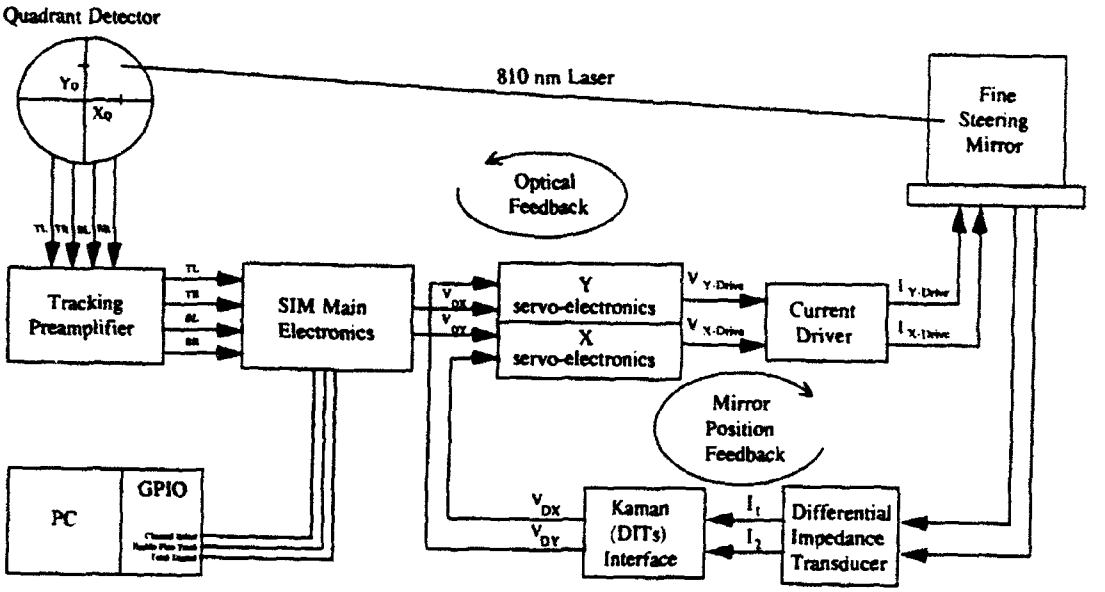


Figure 4. Fine steering functional block diagram.

Coarse Steering Functional Diagram

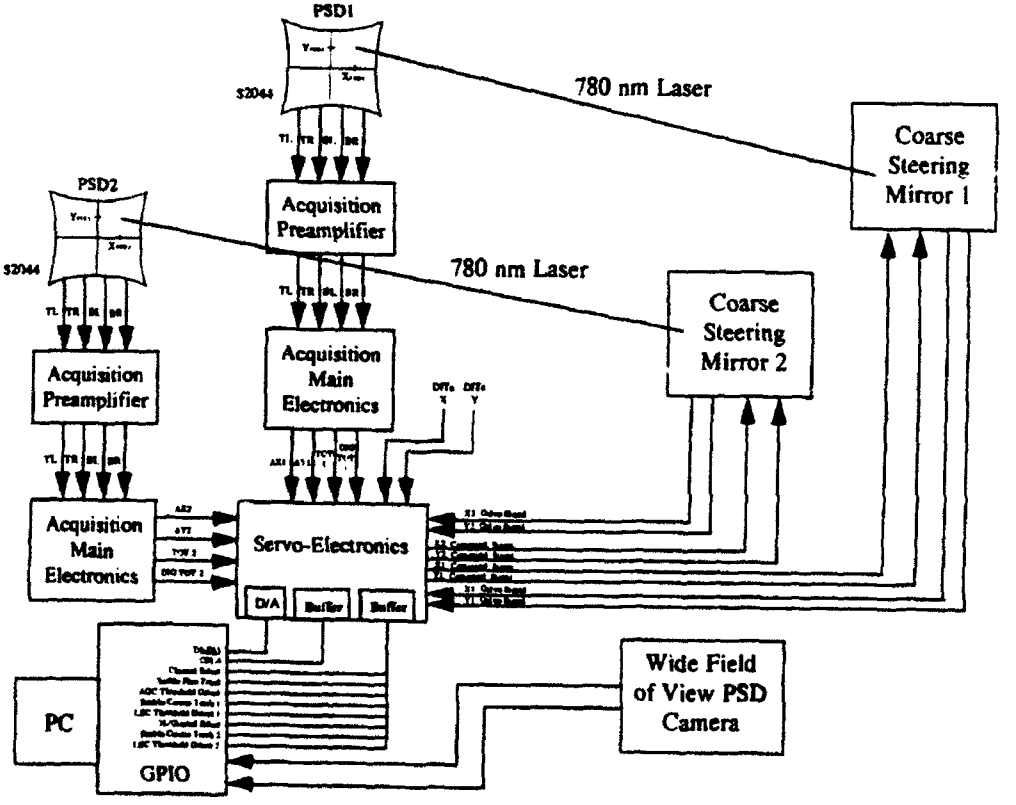


Figure 5. Coarse steering functional block diagram.

Fine Steering Block Diagram

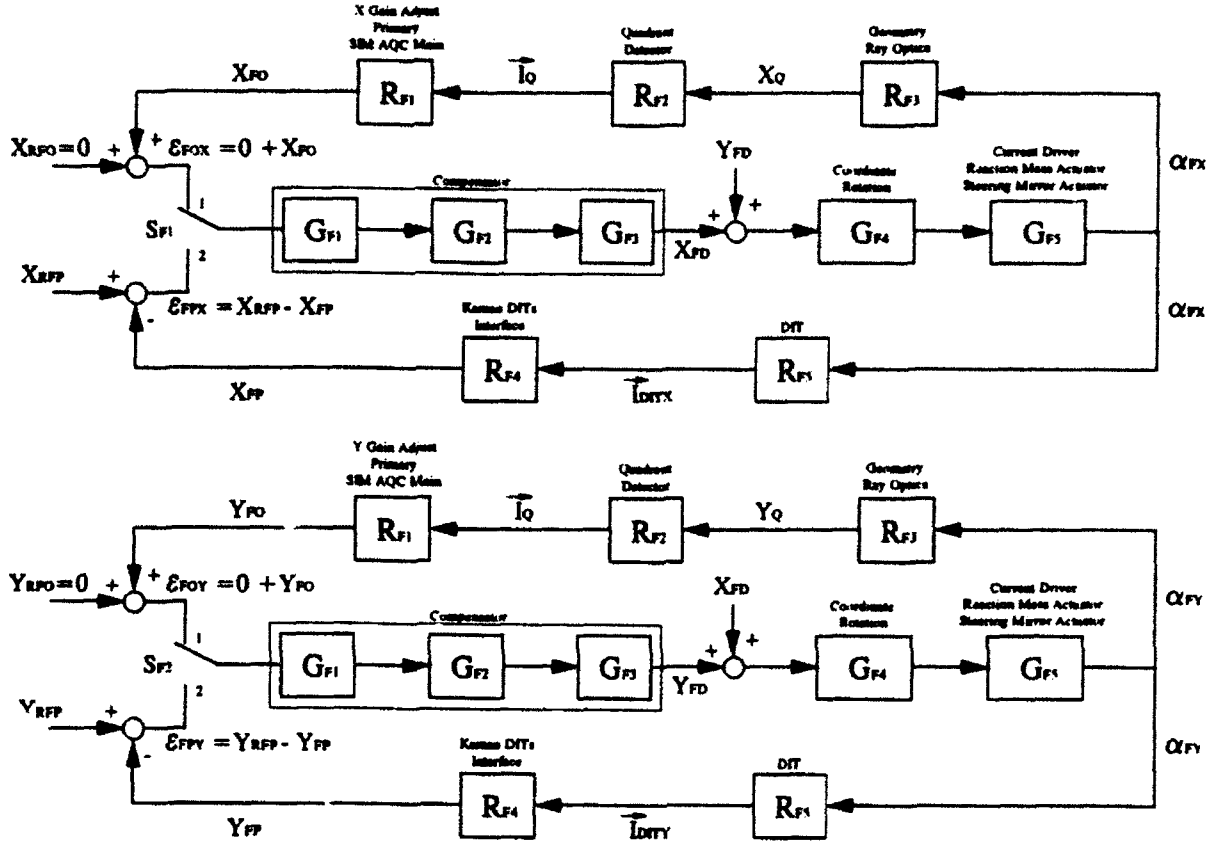


Figure 6. Fine steering control block diagram.

Coarse Steering Block Diagram

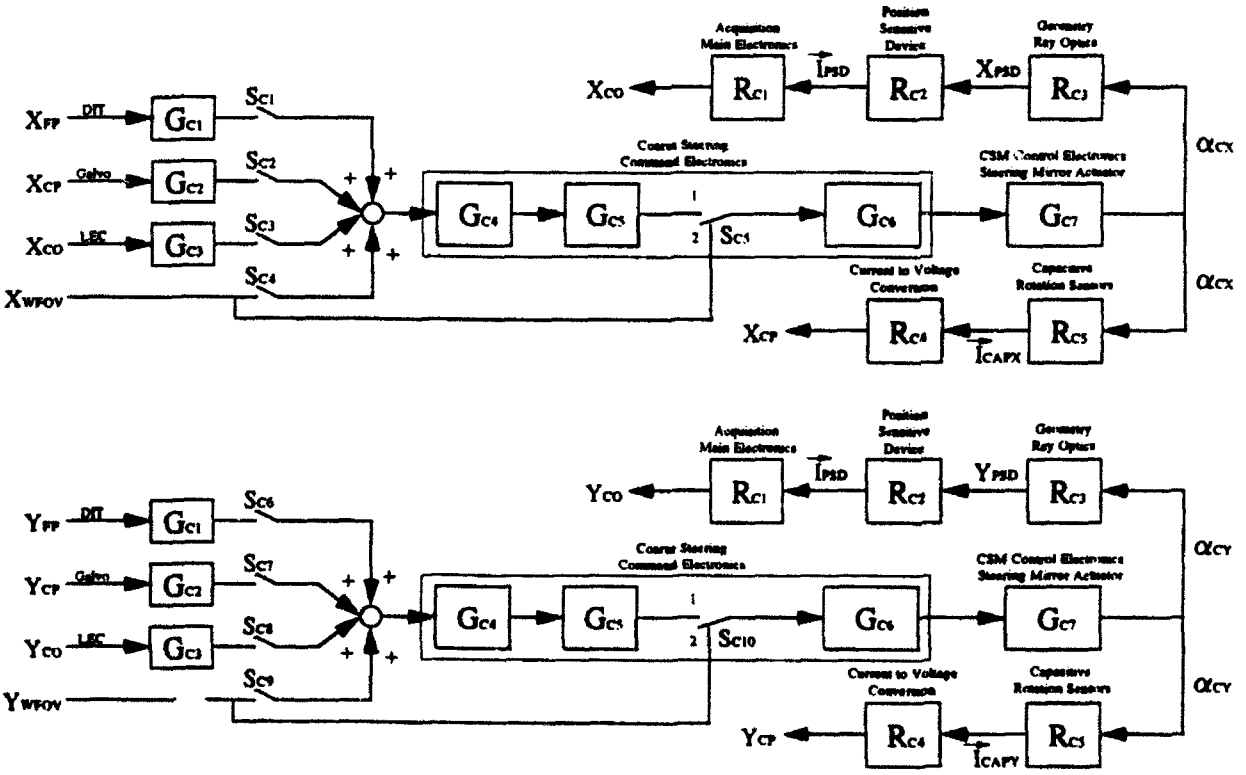


Figure 7. Coarse steering control block diagram.

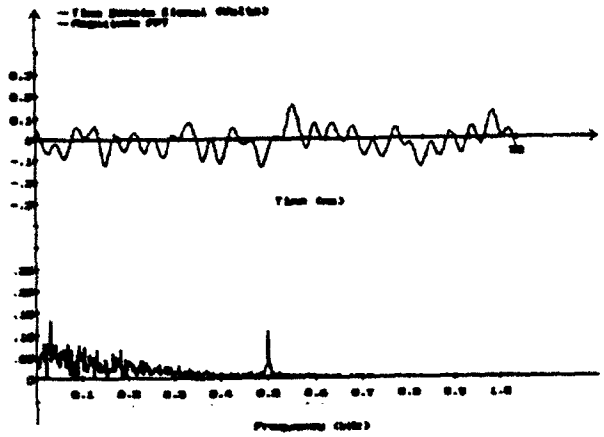


Figure 8. User-defined jitter spectral analysis. The first graph shows the time domain jitter signal while the bottom graph shows the spectral content of the jitter.

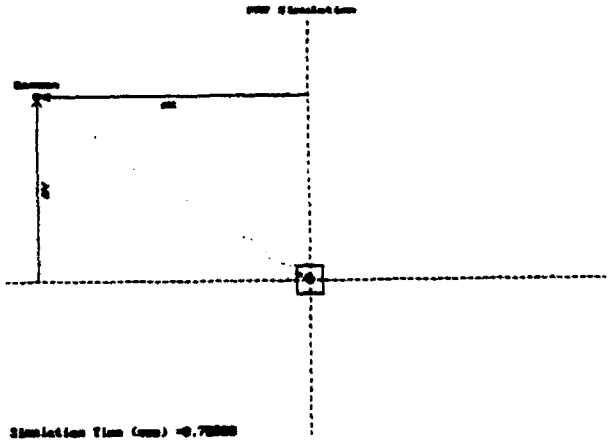


Figure 9. Two dimensional simulation results for wide field of view camera subsystem with jitter.

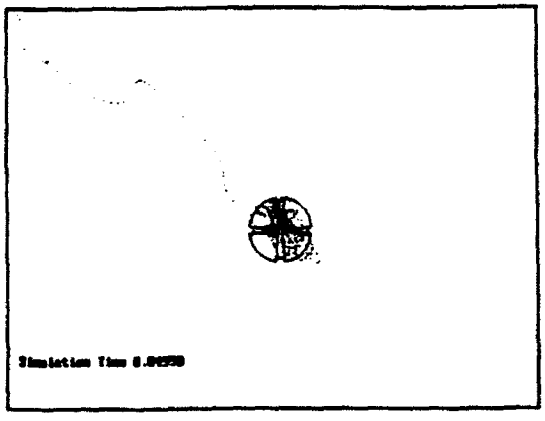


Figure 10. Two dimensional simulation results for coarse steering subsystem with jitter.

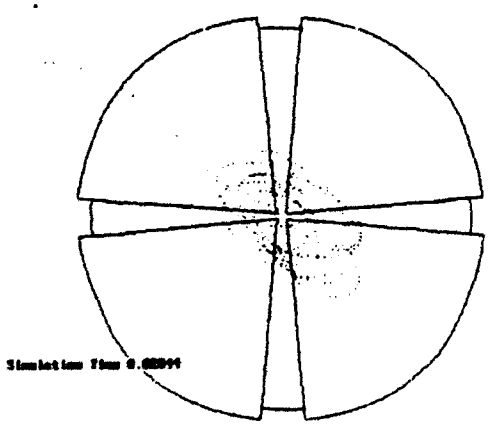


Figure 11. Two dimensional simulation results for fine steering subsystem with jitter.

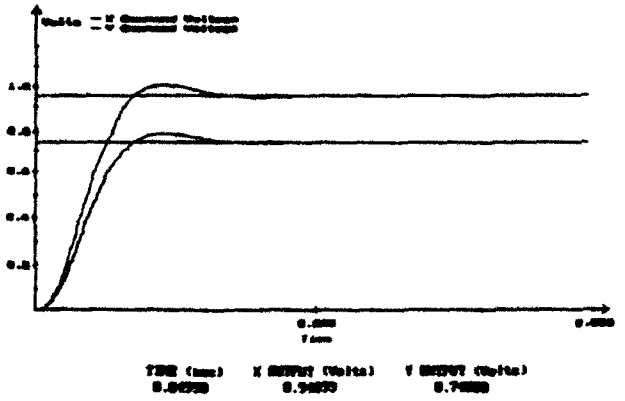


Figure 12. One dimensional simulation results for coarse steering subsystem without jitter.

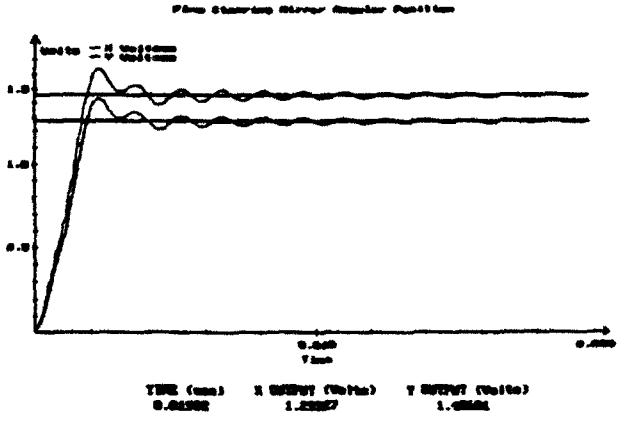


Figure 13. One dimensional simulation results for fine steering subsystem without jitter.

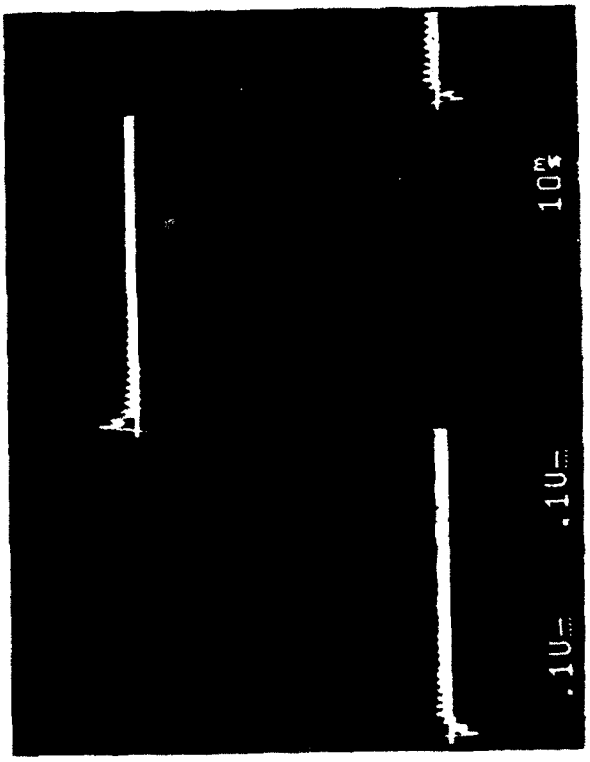


Figure 14. Oscilloscope picture of the fast steering mirror resonse to a step reference signal.

CONGESTION CONTROL FOR ATM NETWORKS IN A TACTICAL THEATER ENVIRONMENT

Benjamin W. Hoe
Department of Electrical Engineering
Polytechnic University

Final Report for:
AFOSR Summer Research Program
Rome Laboratory
Griffiss Air Force Base

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, DC

August 1992

CONGESTION CONTROL FOR ATM NETWORKS IN A TACTICAL THEATER ENVIRONMENT

Benjamin W. Hoe
Department of Electrical Engineering
Polytechnic University

Abstract

Rome Lab is currently developing a next-generation experimental network known as the Secure Survivable Communications Network (SSCN), based on broadband Asynchronous Transfer Mode (ATM) technology. In ATM, all types of information (data, voice, video, message) are placed in 53 byte long packets (ATM cells) and transmitted over available media. In the commercial arena, the primary transmission medium for ATM is fiber, with rates beginning at OC-3 (155.56 Mbps), using the STS-3C SONET protocol. Because of its high transmission speed and switch architecture, congestion control and queue management in ATM networks becomes an important and complex research issue, especially when complicated by the low-throughput, high bit-error transmission links encountered in the military tactical environment. This issue is further complicated in military conditions where traffic patterns are dynamically changed by jamming, degradation of communications resources and security requirements. There is much literature on and many techniques for congestion control in "normal" traffic conditions; however, congestion control in dynamic environments (tactical theater environment) is a poorly documented area in ATM networking. The congestion control and queue management techniques used in the SSCN project should not only be capable of handling traffic at OC-3 rates (155.56 Mbps), but should be able to evolve to support rates in the OC-12 (622 Mbps) range and beyond in the future. First, this paper presents an analysis of congestion control techniques to be used in the SSCN program. Then, potential research issues related to congestion control in a tactical environment are discussed and future research issues are identified.

CONGESTION CONTROL FOR ATM NETWORKS IN A TACTICAL THEATER ENVIRONMENT

Ben W. Hoe
Department of Electrical Engineering
Polytechnic University

1. Introduction

Rome Laboratory is currently developing an experimental wide-area communications network known as the Secure Survivable Communications Network (SSCN) in order to meet the Tactical Communications Land Combat Zone Post-2000 needs for integrated information: voice, data, video, and message. The experience gained from Desert Storm also suggests that future military communications networks must be highly robust and capable to allocate integrated services (voice, data, video) under stressed conditions. The increasing use of high technology such as target identification, pattern recognition using image processing techniques and large computer data file transfers require broader bandwidth than existing communications networks can provide. The ATM concept provides adequate bandwidth, but effective congestion control techniques under stressed conditions are yet to be developed. Most of the queue management and congestion control techniques are designed for commercial applications where the prediction of user traffic patterns is relatively simple and well defined. In military applications, the characteristics of a network are different; it has higher bit error rate due to the enemy jamming, dynamically changing traffic flow patterns and diverse communications media. All these dynamic factors can be categorized as shown in next section, and used in evaluation of congestion control techniques which are being conducted, and are proposed to continue as follow up research under the Research Initiation Program (RIP) sponsored by The Air Force Office of Scientific Research (AFOSR)

[4]. The congestion control technique used in the SSCN program should not only be capable to manage traffic with OC3 speed (155.56 Mbps) but should be able to evolve to OC12 (622 Mbps) and beyond in the future.

2. Congestion control techniques to be used in the SSCN testbed and related research issues

In traditional queue management, output buffering has been shown to be the best when considering delay/throughput performance [5]. GTE corporation has implemented a technique which uses both input and output buffers for congestion control in the SSCN project [3]. Before an ATM cell is processed within this congestion control mechanism, the prioritization is performed by using the contents of a GTE-defined node tag. The node tag is 5 bytes (octets) long and is appended to an ATM cell before it is injected into the ATM switch fabric module. It contains 16 bits for internal routing information, 3 bits for cell loss priority control, 3 bits for hardware priority level, 1 bit for Audit Trail Cell and bits for parity check. The Audit Trail Cell (ATC) flag is used to prevent corrupted routing information from resulting in the misrouting of data. This flag is set upon detection of a parity error. The congestion control technique utilizing both input and output buffers is illustrated in figure-1 and figure-2.

It is designed such that congestion occurs at the output buffers first and is pushed back to the input buffers. The input buffers start to queue up the traffic as the traffic in output buffers exceeds the predetermined threshold. Observe in figure-1, that the traffic at the output queues exceeds the threshold, so an input queue starts queuing up the incoming traffic until the output queues are available for service again.

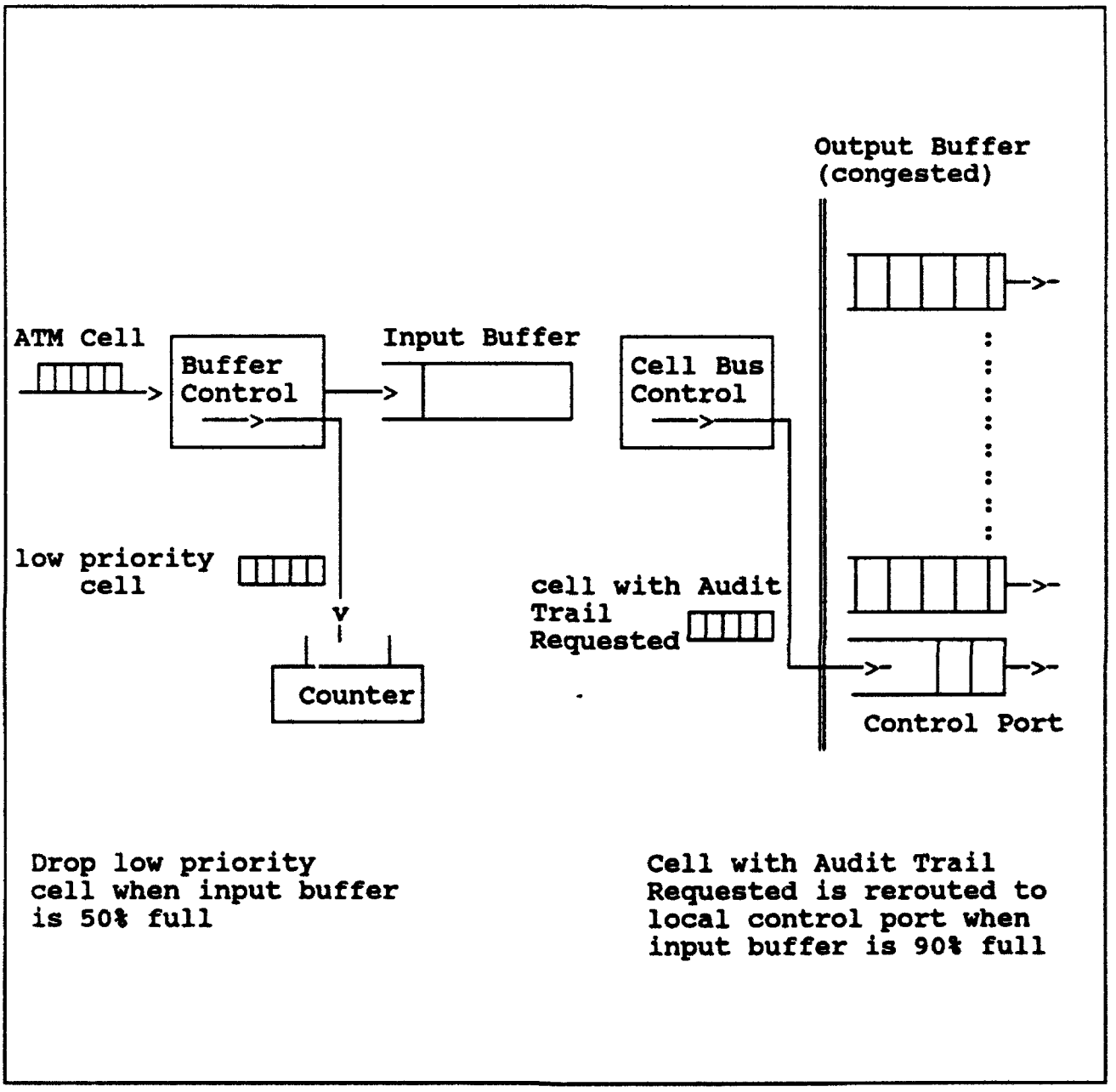


Figure 1. Input buffer response when output buffer is congested

If the congestion at the output queues continues so that the input buffer continues to fill up and exceeds 50% of its queue length, then the input queue will start dropping cells according to their priority level. First, low priority cells are dropped as shown in figure-1 and once input queue is 90% full, the cells with the audit trail bit set will be re-routed to the switch control port. These audit trail bits are used for security reasons and a cell with the audit trail bit set cannot be dropped until it is rerouted to local control module and accounted for [3].

This technique provides temporary congestion avoidance, but more flexible use of the input queues can minimize unnecessary congestion in the input circuit. Consider the following simple example, where all output queues are full except the output queue-1. Because of the back pressure method some input queues start to queue up the traffic until the output queue is once again available. As illustrated in figure-2, the Nth input queue buffers the traffic because the first cell inside the queue is destined to Nth output queue which is congested. However, the second cell in Nth input queue is destined to 1st output queue which is available and in an idle state. But this

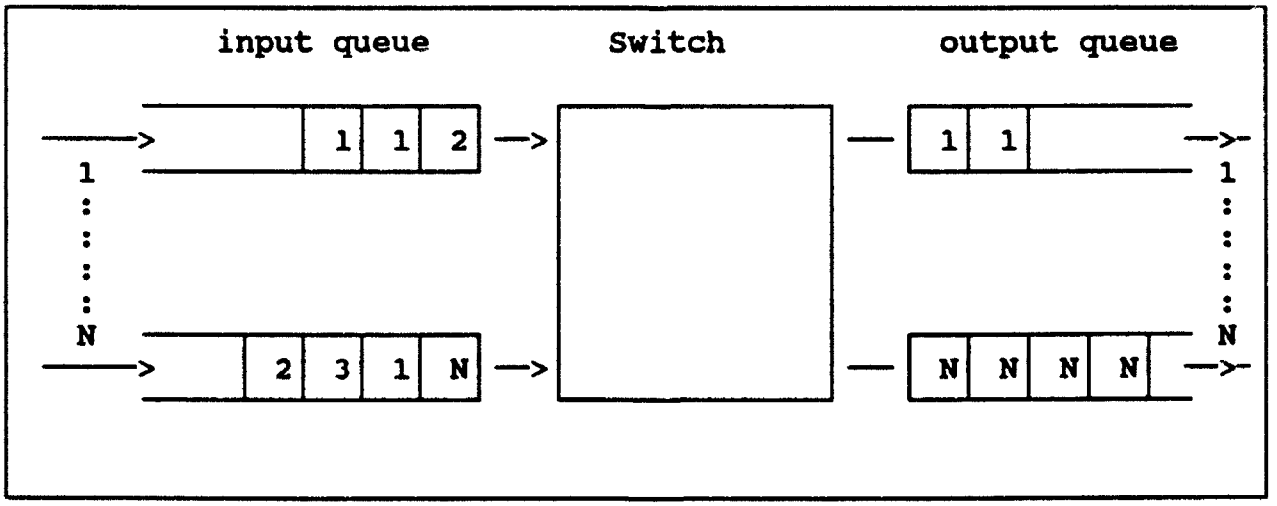


Figure 2. Queue model with input and output buffering

cell cannot go to output-1 until the cell ahead of it is dequeued first. One possible way to solve this problem is by relaxing the FIFO concept and employing an appropriate queuing algorithm at the input buffer.

Another potential research area in queue management is to develop a method to handle long burst traffic. The high bit rate traffic (long burst data) has greater impact on others than the low bit rate traffic. A queue model should have the capability to manage the traffic in the way that high priority cells will be served first and when the congestion level exceeds the threshold, low priority cells will be discarded, and provide equal treatment to same priority cells. Consider the following case where long burst traffic from user-A arrives at the queue first and then low burst traffic from user-B arrives at the queue. We assume that both user-A and user-B have the same priority level. In this case, equal treatment for same priority users is not possible since long burst traffic stream occupies the queue and the server longer than the low burst traffic. Consequently, the mean delay for low burst traffic user-B is higher than that of high burst traffic user-A [1].

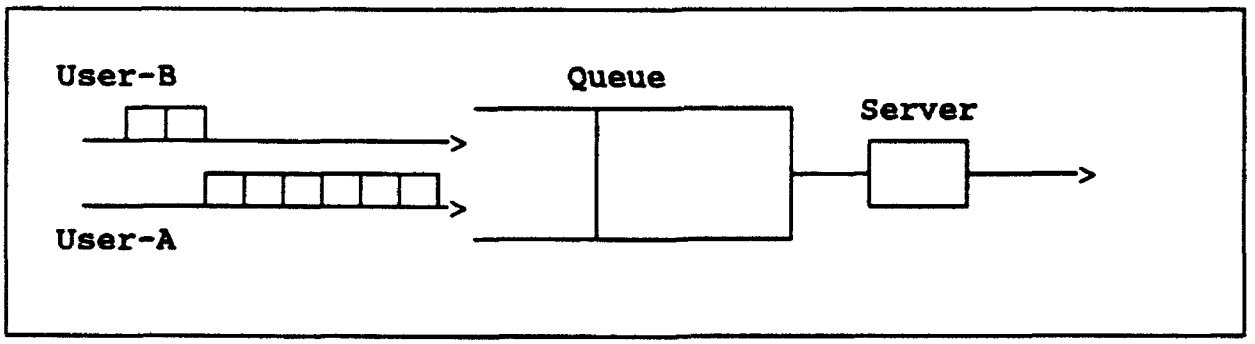


Figure 3. The effect of long burst traffic on a low burst traffic user

This problem can be solved by assigning a departure sequence number based on a concept known as Virtual Clock. It is expected that more long burst traffic will be generated by users in war time scenarios than in normal operation. The study and analysis of bursty traffic and its impact on low data rate users is critical, and utilization of appropriate queuing algorithm pertaining this issue can enhance the performance of the congestion control mechanism under worst case scenarios.

3. Research Effort

The research on congestion control mechanisms for tactical networks was conducted jointly by the author and Rome lab personnel with the technical information obtained from GTE Government Systems. The objective of this research was to identify the potential research issues associated with congestion control techniques in tactical ATM networks, create the simulation model, and perform analysis with different dynamic parameters. As result of this research, we have identified some critical issues which can be classified into the following three major areas:

1) Analysis of diverse traffic classes
2) Analysis of diverse transmission media
3) Threat scenarios

3.1. Analysis of diverse traffic classes

The diverse traffic types must be identified and classified for utilization in simulation and performance analysis of ATM switches in a military environment. The analysis of diverse traffic classes and their impact on tactical ATM networks is

crucial since each type of traffic has different characteristics in terms of bandwidth sensitivity and bit error sensitivity. For example, video signals are delay sensitive, where long delays of video signals will result in unrecognizable images at the receiver end. However, one bit error on video is acceptable since it will not degrade the quality of the image significantly. On the other hand, data is very sensitive to bit errors; one bit error can result in a completely different expression at the receiver end. The traffic arrival pattern is another critical factor in the tactical environment. The traffic flow in peace time (normal operational mode) is similar to the traffic patterns in commercial networks with some additional security features. However, traffic patterns in a hostile environment will be different. In hostile operational modes, the traffic loading on all links will be changing dynamically. Some links or channels will be destroyed or degraded by enemy attacks and bottleneck effects are expected on remaining links.

In order to thoroughly understand different traffic classes and their impact on congestion control, the simulation of different traffic patterns with different behaviors is required, using the node model to be implemented in the SSCN testbed. The study of different traffic patterns has been initialized in Rome Lab as part of the summer research effort, but more detailed analysis and evaluation of diverse traffic classes remains to be done, because of the time limitation and administrative delay in technical information transfers from GTE Government Systems.

3.2. Analysis of diverse transmission media

The OPNET tool provides three generic link models that can be used to characterize the type of connectivity that a particular transmission medium utilizes: a point-to-point link,

a bus link and a radio link are available options. These three types of link connectivity models will be analyzed in terms of their respective parameters. For example, in a point-to-point link, transmission delay, propagation delay and bit error rate will be defined and analyzed. Transmission delay and propagation delay characterizes the transmission speed, and bit error rate indicates how much a link is corrupted with errors. It is known that higher bit error rates will be encountered in tactical network than in commercial networks. The error threshold is used to indicate the level of threshold that is allowed on particular link. If a link (or) a channel is corrupted with noise so that its error rate exceeds the error threshold, then this particular link should be taken out of service; error detection and correction effort on cells arriving from such a link would be difficult if the error rate is very high. In bus links, packet collision and closure factor will also be considered in addition to the delay and bit error factors considered in point to point links. It is critical to consider collision because packet collision in bus topologies is unavoidable when users can transmit packets to the same destination at the same time on the same medium. The closure factor is also an important consideration, since it indicates the eligibility of connectivity between transmitter and receiver (e.g., for security reasons). In a radio link, more factors such as signal to noise ratio (SNR), and background noise (thermal noise) must be included in the analysis. The focus of this research was on point-to-point and bus type links, since these two types will be used in the SSCN initial five node testbed, while radio links (which require detailed link model definitions) should be added in the future when the models of interest have been selected for development in the SSCN program. Further study on transmission media and their impact on congestion control is recommended to help characterize the behavior of dynamic tactical networks.

3.3 Threat Scenarios

Military threat scenarios have been documented and implemented in a network management system prototype testbed under the Communications Network Operating System II (CNOS II) project conducted for Rome Lab by Stanford Telecommunications, Inc. The CNOS II program developed an IMS (Integrated communications network Management System) testbed containing extensive features to define various threat parameters, and model the response of the IMS to various user-defined network models and dynamic scenarios of interest. The functionality of the IMS and its role in tactical ATM networks is discussed in the next section. As part of the research effort, threat scenarios were identified as one of the critical issues in tactical ATM networks. The thorough study of threat scenario models available in the IMS prototype testbed, and the adaptation of those models to support the SSCN project is recommended. Any congestion control mechanisms to be used in a tactical theater environment must be evaluated under different threat scenarios to ensure their performance under worst case conditions. The main component of the SSCN project is a Multi-Level Secure (MLS) Tactical Switch (MTS) node, or ATM gateway switch, the buffering mechanisms for which was designed using the OPNET design tool of MIL3, Inc.. Therefore, it is recommended that threat scenarios be defined using OPNET to evaluate the MTS node (ATM switch), which is already modelled on OPNET. Our research reveals no significant work has been done on defining threat scenarios using the OPNET tool, hence, the author and Rome Lab identify the modeling of threat scenarios on OPNET as potential future research work.

4. The Integrated Communications Network Management System (IMS) and its role in the SSCN

The Integrated Communications Network Management System (IMS) is a next-generation network management system which was developed under the Communications Network Operating System II (CNOS II) project. The CNOS II project was a follow-on to the CNOS I project, which was initiated by Rome Lab to define the high level architecture for a highly robust and survivable communications system that will integrate multiple networks and transmission media. The IMS prototype was developed under the CNOS II project to evaluate the efficiency and effectiveness of network management algorithms in different dynamic scenarios of interest. The IMS architecture is briefly discussed in this section because of it's crucial role in SSCN project. As illustrated in figure-4, network management processing functions were partitioned into four management subsystems: Service Manager, Administrative Manager, Facilities Manger, and Resource Manger. Each manager facilitates certain functionality. The Service Manager handles user interfaces, access control and mapping of requested services to transmission resources. The Resource Manager handles transmission system interfaces, and monitors specific transmission links and controls their operation, including connection establishment. The Facilities Manager is responsible for monitoring overall performance and access control to existing networks, and all available end-to-end transmission resources. The Administrative Manager manages non-real time functions associated with user access priorities, privileges and security level as well as configuration information [7]. In addition to these four network management processing subsystems, there is an information processing subsystem which contains the Management Information Base (MIB) and Management Transfer Part (MTP) [7][8]. The network management decision making algorithms reside in the management

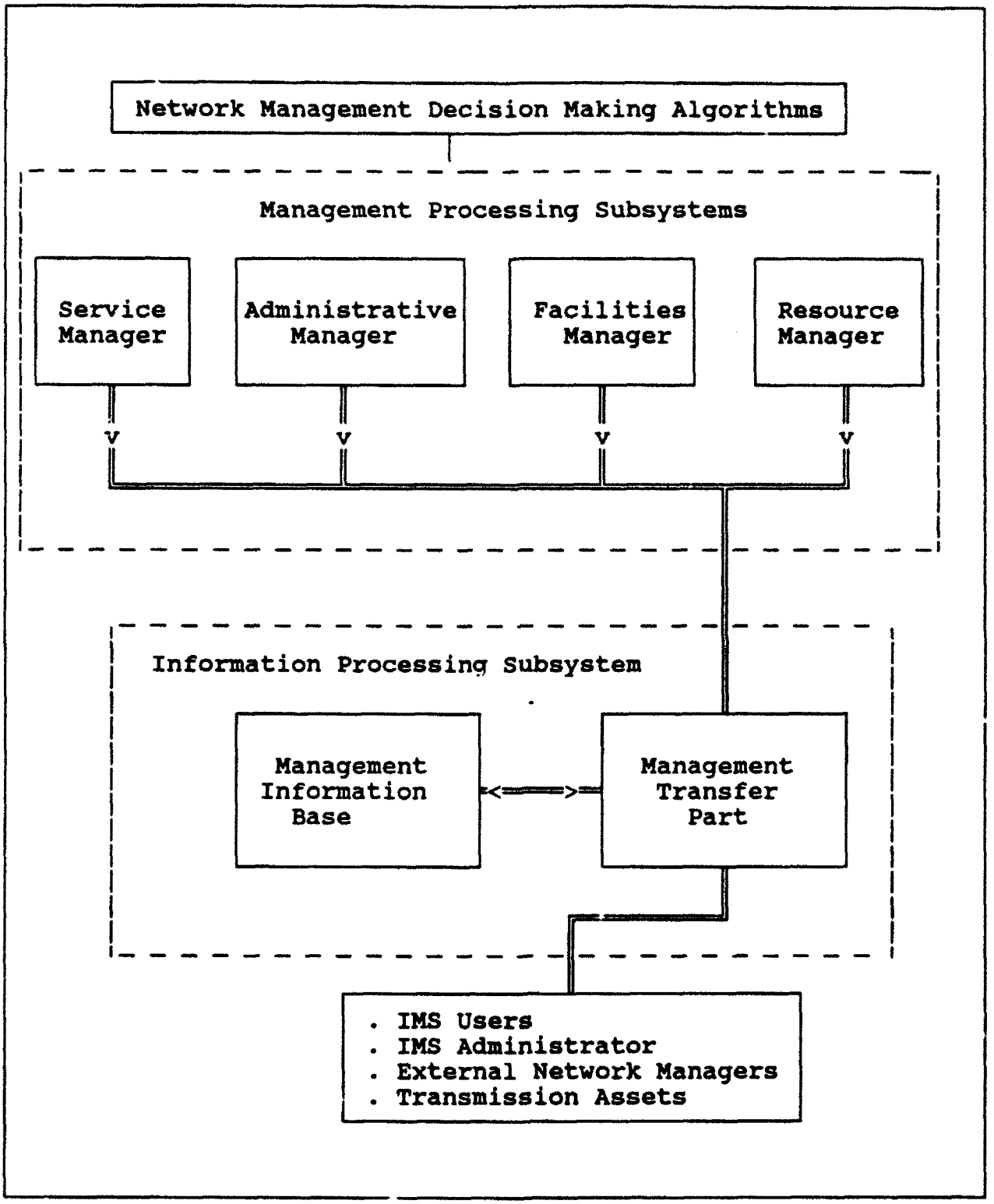


Figure-4 Top level IMS architecture

processing subsystem. Each IMS node (domain) develops a nodal viewpoint (local knowledge) of the network state and, through coordination with other IMS nodes, attempts global user service optimization. A three node IMS prototype was developed under the CNOS II project, which contains the aforementioned network management functionality, and operates on simulated traffic and network resources. This prototype allows a user to define various parameters associated with a tactical theater environment such as Mean Time Between Failure (MTBF), Mean Time To Repair (MTTR), Bit Error Rate (BER), and more specific conditions such as a particular link being disabled at certain times, due to enemy jamming, and operational at other times. Such threat situation can be defined in generic classes: link, node, network, equipment threats, or more specific events: a node or a link will be taken out of service at specific time, etc. These threat parameters can be used in the analysis of ATM Switch performance to evaluate how well it performs under stressed conditions. The IMS plays a important role in the SSCN project; not only by its ability to evaluate the performance of a network in dynamic conditions, but it is, in fact, able to control the tactical ATM switches in response to those adverse conditions. The effectiveness of congestion control mechanisms in an ATM network also depends on how effectively the IMS can optimize the mapping of service requests onto available transmission media. For example, the IMS must be able to allocate resources to services requested so that congestion at a single node (or a group of nodes) can be avoided.

5. Discussion

The objective of this work is to identify the potential research problems pertaining congestion control in tactical ATM networks. As result of the research, we have identified some

critical issues, which were presented in previous sections. More detail and a thorough investigation of these issues against potential queue management techniques, either proposed by GTE or identified by Rome Lab and the author is recommended. The buffering mechanisms in the MTS ATM switch to be used in SSCN project were designed with the OPNET design tool. The OPNET tool utilizes object oriented software, and allows users to model a network hierarchically. Network analysis can be performed at different levels without having to worry about the higher or lower level in OPNET. Therefore, OPNET was selected as our analysis tool in this research and for a follow-up research effort. One major task still remaining to be carried out is defining threat scenarios on OPNET simulation. OPNET does not provide direct ways to define the threat scenarios, as the IMS testbed does, but this research revealed that it is possible to define threat scenarios in terms of link attributes, "C" code instructions in a State Machine, and by using different node attributes and traffic patterns. The detailed and complete construction of threat definitions on OPNET is a time consuming process and is still yet to be conducted.

6. Conclusion

This research revealed that further work in congestion control for tactical ATM networks is required. All congestion control techniques to be used in the SSCN testbed must be thoroughly evaluated in a dynamic environment to validate the expected performance of the current design in achieving the best possible performance in a full scale deployment of the future SSCN ATM switches. In fact, this research is an important component of a much larger military initiative to implement a more global (multiple government & commercial networks, multiple

vendors, etc.) ATM environment in which congestion control will be much more complicated by the size of the networks, their speed and switch architectures.

Acknowledgement

I would like to express my gratitude to Nick Kowalchuk, Rome Lab, for reviewing this paper and for valuable suggestion regarding SSCN project. His guideline and comments have great influence on my research as well as on this paper. I would also like to thank my lab focal point and branch chief Mr. Joseph Zdanowicz for his help and assistant. His concern on not only my research project but my future educational plan is greatly appreciated. I am also grateful to other staff members of Communications Network Branch for their assistant during this research project.

References:

1) H. J. Chao,"A Novel Architecture for Queue Management in the ATM Network," IEEE Journal on Selected Areas in Communications., vol. 9, No. 7, pp.1110-1118, Sep. 1991
2) GTE Government Systems, "Secure Survivable Communications Network, " User Meeting Report, June 8-9, June 15-16, 1992.
3) GTE Government Systems, "Secure Survivable Communications Network," Technical Proposal Vol. III, March 1992
4) B. W. Hoe,"Congestion Control for ATM Network in a Tactical Theater Environment", Research Proposal for AFOSR sponsors Research Initiative Program, August 1992.
5) M. J. Karol, M. G. Hluchyj, and S. P. Morgan, "Input versus output queuing on space-division packet switch, " IEEE Transaction on Communications, vol. 35, no. 12, pp. 1347-1356, Dec. 1987.
6) R. Schmidt, M. Fuglestad, L. Laws, and J. Ormord, "A prioritized ATM switch node with hardware priority and software bandwidth allocation for a high speed Integrated network," IEEE C^3I Conference, pp. 367-371, June 1992.
7) Stanford Telecommunications, Inc., "The Communications Network Operating System (CNOS) II," Final Review, March 1992.
8) Stanford Telecommunications, Inc., "Software User's Manual for The IMS Exploratory System Model of the CONOS II Program," F30602-89-C-0203, March 1992.

THIS PAGE INTENTIONALLY LEFT BLANK

MAXIMUM LIKELIHOOD BASED IMAGING OF PRECESSING RADAR TARGETS

Kenneth E. Krause
Doctoral Student
Department of Electrical Engineering

Washington University
Campus Box 1161
One Brookings Drive
St. Louis, MO 63130-4899

Final Report for:
Summer Research Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D.C.

August 1992

MAXIMUM LIKELIHOOD BASED IMAGING OF PRECESSING RADAR TARGETS

Kenneth E. Krause
Doctoral Student
Department of Electrical Engineering
Washington University

Abstract

The Maximum Likelihood approach is described and applied to the imaging of radar targets undergoing cooperative precessional motion under narrowband millimeter wave radar signal illumination. A comparison with conventional processing is made when simulated Gaussian noise is added to the measured data. With the maximum likelihood algorithm, useful images are generated from relatively small amounts of noisy data. The measured data used in the image reconstructions were measured at the Rome Laboratory Prospect Hill Facility in Waltham, MA.

MAXIMUM LIKELIHOOD BASED IMAGING OF PRECESSING RADAR TARGETS

Kenneth E. Krause

1 Introduction

Radar images of targets are formed by processing the echo data received over an appropriate angular aperture and frequency bandwidth. When a rotation about the axis normal to the radar line-of-sight is used in conjunction with a wide radar bandwidth, conventional inverse synthetic aperture (ISAR) radar images can be obtained [1]. Images can also be obtained from angular motion in two directions, such as in the 'angle-angle' imaging described by Brown[2].

The image reconstructions in these and similar situations are obtained by performing a Fourier transformation on the received data. This approach is based on the theoretical relationship between the received signal and the reflectance image when the target consists of independent scatterers.

The above described processing makes no provision for noise in the data. The work described in this report describes imaging from a perspective that accounts for additive noise in the measured radar data. Specifically, based on the assumption that the target consists of independent scatterers, an imaging approach based on maximum likelihood estimation theory is used to estimate the reflectance image of a target when the measured data is contaminated by additive Gaussian noise. Images so obtained are compared with images obtained by conventional processing. For the processing used in this report, the reasonable approximation is made that the scattering targets are planar.

The radar data measurements were obtained from the Rome Laboratory Prospect Hill Facility located in Waltham, MA. The system there collects imaging data using a stabilized, narrowband, 140 Ghz transmitted signal. The stabilized 140 Ghz is obtained using a Gunn diode oscillator at 46.7 Ghz followed by a frequency tripler and a specially designed electrical and mechanical feedback system. See [3] and [4] for details on the hardware.

Section 2 describes a conventional processing formulation for imaging that was developed by Rome Laboratory. Next, Section 3 describes the maximum likelihood approach the author implemented in his summer work assignment at Rome Laboratory. Section 4 is the summary.

2 Conventional Processing

The geometrical configuration for the radar measurements and derivation of the reconstruction algorithm are described in [5]. This section summarizes from [5] the basic measurement equation and the notation to be used in this report. Figure 1 shows the measurement configuration. The cooperative target motion utilized for imaging is now described. The target is located in the xy-plane with rectangular coordinates $(x, y, 0)$ and polar coordinates $(r, \psi, 0)$ given by $x = r\sin\psi$ and $y = -r\cos\psi$. We define the positive p-axis as lying in the xy-plane oriented at angle θ measured from the negative y-axis. At each p-axis angle θ, the target is tilted by angle α about the p-axis, and a coherent radar measurement made. The p-axis steps in θ for a fixed tilt angle α, as defined above, for each radar measurement. The set of data collected as θ varies for a fixed tilt angle α is called a ring of data.

Based on this measurement and geometry, the following equation is deter-

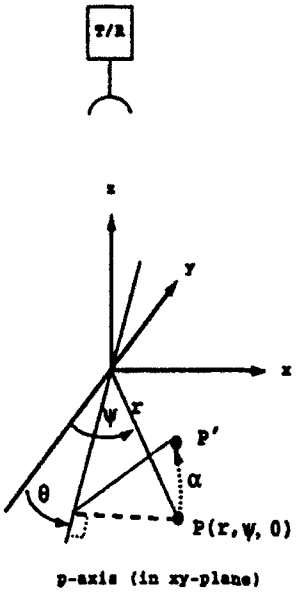


Figure 1: Radar measurement for precessing target.

mined for the image $g_\alpha(r,\psi)$ in terms of ideal measurements, $G_\alpha(\theta)$.

$$g_\alpha(r,\psi) = \frac{2\sin\alpha}{\lambda}\int_0^{2\pi} G_\alpha(\theta)e^{-j4\pi r\frac{\sin\alpha}{\lambda}\sin(\theta-\psi)}d\theta. \tag{1}$$

The subscript α denotes that the image is due to the ring of ideal data $G_\alpha(\theta)$. The resolution obtained in this case is $0.4\lambda/\sin\alpha$, the width of the central lobe of the point spread function between zeros. The final image is obtained by adding several such single-ring images. The following is the equation for the final image in terms of single-ring images.

$$g(r,\psi) = \sum_\alpha g_\alpha(r,\psi) \tag{2}$$

The received data are given by the expression

$$G_\alpha(\theta) = e^{-j4\pi\frac{\sin\alpha}{\lambda}(x_m\cos\theta + y_m\sin\theta)} \tag{3}$$

for an ideal point target of unit strength located at coordinates $(x_m, y_m, 0)$.

Figure 2 provides simulation results that show how the image from an ideal

Figure 2: Point Target-Conventional Processing: a. without noise, b. with noise.

point target varies when noise is added to the data. The image reconstructed from data corresponding to an ideal point target of unit strength located at coordinates $(15, -5, 0)$ is shown in figure 2a. Figure 2b shows the reconstruction from the same signal data as the previous case, but with Gaussian noise added. Simulated point target data had unit magnitude. The variance of the simulated Gaussian noise added to the simulated data was 4.0. The xy-plane reference location in the three-dimensional plots is $z = 0$. It is shown elevated for clarity of viewing. The signal data used for this simulation corresponds to a tilt angle α of 10^o. Increasing non-target structure is seen as noise is introduced into the signal data.

Images obtained in [5] were formed by adding four rings of data with angles α corresponding to equal increments of $\sin\alpha$ and $\alpha_{max} = 10^o$. Images in [6] were obtained using 16 rings of data.

The rest of this report will describe an approach to single ring image recon-

struction that explicitly accounts for the noise in the measured data.

3 Maximum Likelihood Processing

Equation (3) will be modified slightly for interpretation in terms of a Maximum Likelihood imaging approach the author has formulated as part of his doctoral research.

The complex envelope representation for the return from a target of reflectance strength B_m at location $(x_m, y_m, 0)$ is given by

$$\tilde{s}_{m\alpha}(\theta) = \sqrt{E_t} B_m e^{j\theta_m} e^{-j4\pi \frac{\mu_{\mathrm{eff}}}{\lambda}(x_m \cos\theta + y_m \sin\theta)} \tag{4}$$

where $\sqrt{E_t}$ is a constant related to the transmitted signal energy of the measurement. This representation of the reflectance by a constant parameter B_m is valid over small variations in aspect angle [7]. The angle θ_m is a random variable, uniformly distributed on $[-\pi, \pi]$, representing the uncertainty in target location on the scale of a wavelength of the illuminating radiation. As before, α is the tilt angle. Scatterer phases will be denoteded by lower case subscripts. The variable θ refers to the p-axis angle. A specific measurement will be referenced by upper case subscripts consisting of numerals or upper case letters.

The received data are then described by

$$\tilde{r}_\alpha(\theta) = \tilde{s}_{m\alpha}(\theta) + \tilde{w}(\theta) \qquad \theta = \theta_1, \theta_2, ..., \theta_N \tag{5}$$

where $\tilde{w}(\theta)$ is complex white Gaussian noise of spectral height N_0. The author has proposed a specific form of the likelihood function to use in an estimation based approach to imaging targets composed of independent scatterers, in a noisy measurement environment. In the present context, for single ring images,

this likelihood function is

$$\Lambda[\tilde{r}_\alpha(\theta); B_m] = \frac{1}{2\pi}\int_{-\pi}^{\pi} d\theta_m \left[\exp[-\frac{1}{N_o}\int_0^{2\pi} \mid \tilde{s}_{m\alpha}(\theta) \mid^2 d\theta] \times \exp[-\frac{2}{N_o} Re(\int_0^{2\pi} \tilde{r}_\alpha(\theta)\tilde{s}^*_{m\alpha}(\theta)d\theta)\right] \quad (6)$$

The procedure is to find the B_m which maximizes this expression for every point $(x_m, y_m, 0)$ in the target plane. At Rome Laboratory, Target Characterization Branch, the solution to this problem was implemented in FORTRAN on a Vax 750 computer. (Earlier study of the algorithm had been done in turbo-Pascal on a Macintosh II computer.)

Carrying out the details, see [8], and ignoring multiplicative constants that do not depend on the parameter to be estimated, (6) becomes:

$$\Lambda[\tilde{r}_\alpha(\theta); B_m] = \exp[-\frac{B_m^2 2\pi E_t}{N_o}] \times \exp[\frac{2}{N_o}\sqrt{E_t} B_m (L_{cm\alpha}\cos\theta_m - L_{sm\alpha}\sin\theta_m)] \quad (7)$$

where

$$L_{cm\alpha} = \int_0^{2\pi} (r_{\alpha i}(\theta)\cos\phi_{\alpha m}(\theta) - r_{\alpha q}(\theta)\sin\phi_{\alpha m}(\theta))d\theta \quad (8)$$

$$L_{sm\alpha} = -\int_0^{2\pi} (r_{\alpha i}(\theta)\sin\phi_{\alpha m}(\theta) + r_{\alpha q}(\theta)\cos\phi_{\alpha m}(\theta))d\theta \quad (9)$$

$$\phi_{\alpha m}(\theta) \equiv 4\pi\frac{sin\alpha}{\lambda}(x_m\cos\theta + y_m\sin\theta) \quad (10)$$

and

$$\tilde{r}_\alpha(\theta) \equiv r_{\alpha i}(\theta) + r_{\alpha q}(\theta) \quad (11)$$

describe the likelihood in terms of all variables defined so far. Defining a scaled reflectance B'_m by the expression

$$B'_m = \sqrt{2\pi} B_m \tag{12}$$

the likelihood expression (7) becomes

$$\begin{aligned} \Lambda[\bar{r}_\alpha(\theta); B'_m] &= \exp[-\frac{B'^2_m E_t}{N_o}] \\ &\quad \times \exp[\frac{2}{N_o}\sqrt{E_t} B'_m (L'_{cm\alpha}\cos\theta_m - L'_{sm\alpha} sin\theta_m)], \end{aligned} \tag{13}$$

with

$$L'_{cm\alpha} = \frac{L_{cm\alpha}}{\sqrt{2\pi}} \tag{14}$$

and

$$L'_{sm\alpha} = \frac{L_{sm\alpha}}{\sqrt{2\pi}}. \tag{15}$$

Equation (13) is the form of the likelihood the author's algorithm requires in order to obtain the maximizing solution B'_m in terms of the variables mentioned above. Once B'_m has been obtained, by equation (12), the solution for refectance B_m at coordinates $(x_m, y_m, 0)$ is easily obtained. This is the maximum likelihood solution. Specific features of the maximum likelihood algorithm are a thresholding operation and solution of the nonlinear equation obtained by setting the derivative of (13) equal to zero. These features cause either $B'_m = 0$ to be the solution, or else a nonzero solution for B'_m is obtained from the nonlinear equation solution. In either case, this results in a conservative(low) estimate of B_m, compared to conventional processing. Figure 3 shows the maximum likelihood algorithm applied to the same data used to reconstruct the conventional images of figure 2. Notice that in the noiseless case, the maximum likelihood estimate is the conventional estimate.(For computational purposes, a low noise

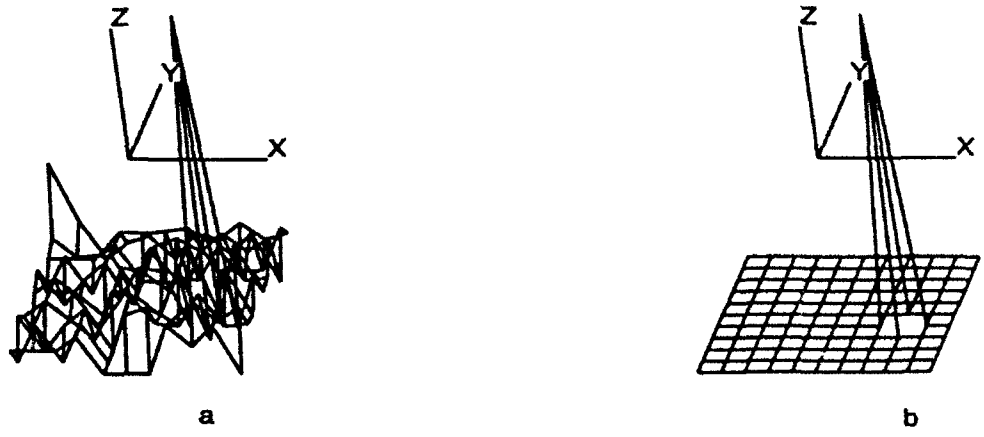


Figure 3: Point Target-ML Processing: a. without noise, b. with noise.

variance was used as input to drive the algorithm for zero noise case calculations). With noise present, the algorithm prescribes the threshoiding operation and calculation of any nonzero reflectance estimates.

Two cases using measured data will now be presented. The measured data from Prospect Hill was normalized so the maximum data point magnitude would be unity. Noisy data was simulated by adding zero mean Gaussian noise of variance 0.1. For plotting purposes, where maximum likelihood estimates are presented on contour plots, $B_m = 0$ solutions are set equal to the minimum non-zero value, 0.0001.

One configuration studied consisted of three spheres with diameters in inches of 0.4686, 0.2186, and 0.25; located at coordinates $(0, 12.15, 0)$, $(-4.45, -2.57, 0)$, and $(7.28, -4.21, 0)$, respectively. Coordinates given are in units of the wavelength of the illuminating radiation, which was 2.14 mm.

Two realizations of noisy ring 16 images are given for this target in figures

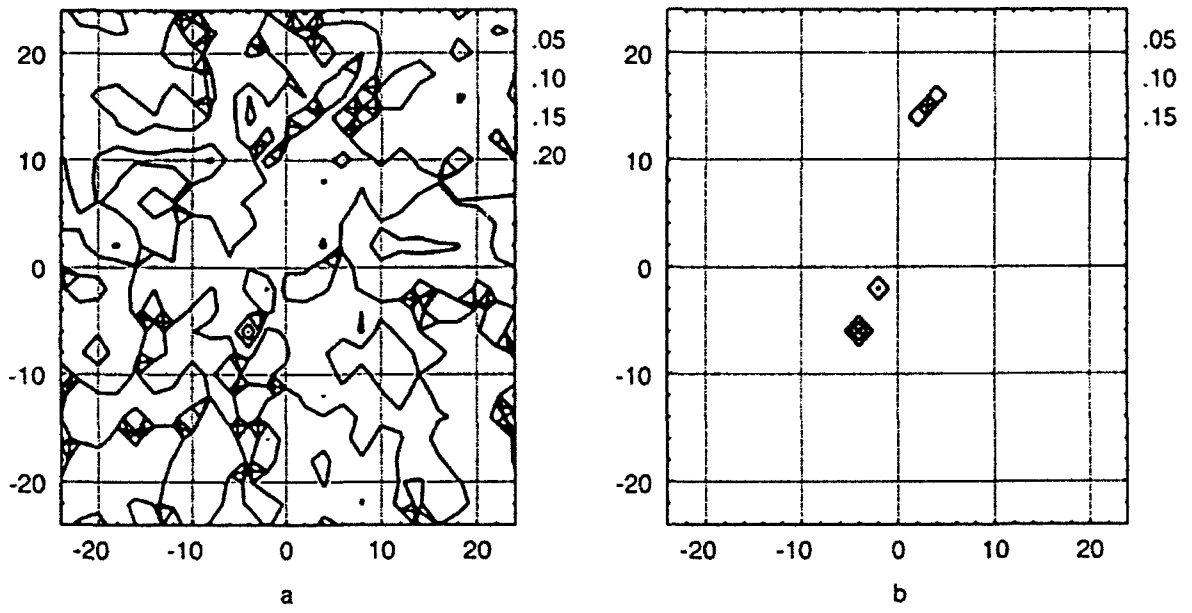


Figure 4: Realization 1: Noisy Ring 16 Image of 3 spheres using a. Conventional, b. Maximum Likelihood Processing.

4 and 5. Each figure contains the data computed directly from conventional and maximum likelihood processing. It is emphasized, for assistance in interpretation, that the contour plot scales are associated with different colors that do not show up on this reproduction. Thus, although the conventional imaging generally finds the same high level image points as the maximum likelihood, it usually finds a few more higher level image points and many more lower level image points. This just means that the image uncertainty implied by the conventional processing is not as bad as the visual appearance implied by the black and white display. Even with this consideration, the maximum likelihood approach does leave fewer extraneous image points and consequently a computed image having less apparent uncertainty.

In the sense that the major geometrical features of the sphere target can be seen in some of the single ring images described above, the sphere target seems to be well modeled by the independent point scatterer models which underly both

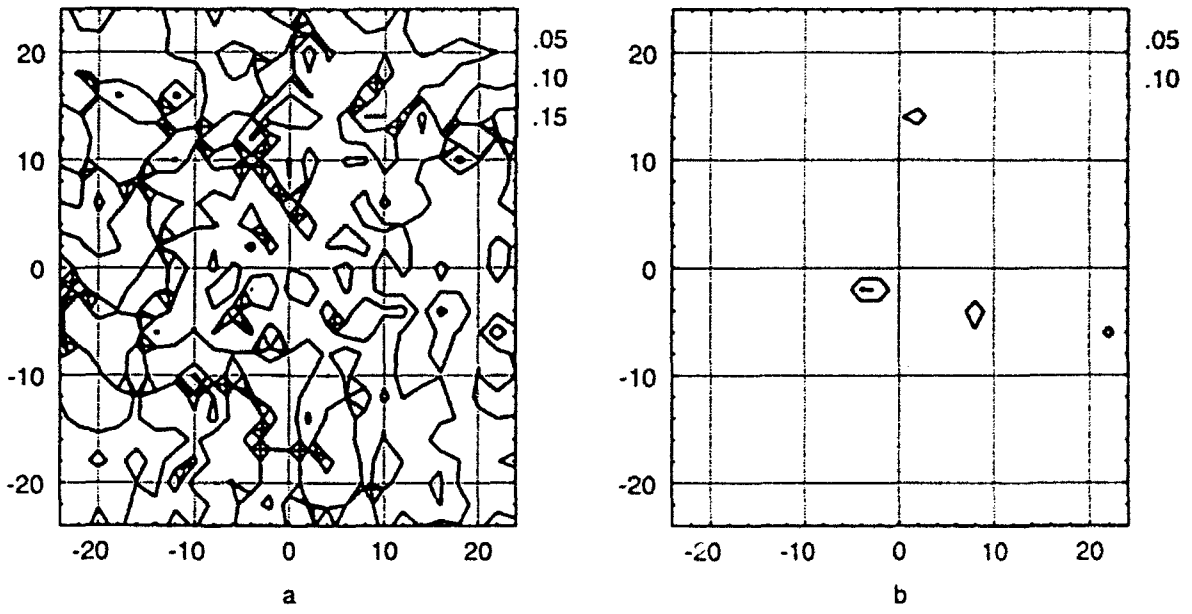


Figure 5: Realization 2: Noisy Ring 16 Image of 3 spheres using a. Conventional, b. Maximum Likelihood Processing.

the conventional and maximum likelihood processing. This is not necessarily the case with all targets studied.

Consider the "RL" target. This is a thin aluminum single structure 83 mm high by 63 mm wide. Its shape is derived from the form of the solid letters "R" and "L", with the "L" part to the right, slightly lower than, and intersecting the "R" part. A conventionally reconstructed and thresholded image, based on 8 rings of data having maximum tilt angle α of about 5^{o} is shown in figure 6a. This thresholded image gives the pictorial idea of the target's structure. Figure 6b shows the image reconstructed from ring 8 of data. The two image scales are different. Looking at the images, it can be seen that all of the recognizable features in the 8 ring reconstruction are not clearly visible in the single ring reconstruction. This is in contrast with the three sphere target. This difference may be due to either level differences of point scatterers under the assumed model or to more complicated scatterering mechanisms requiring a different

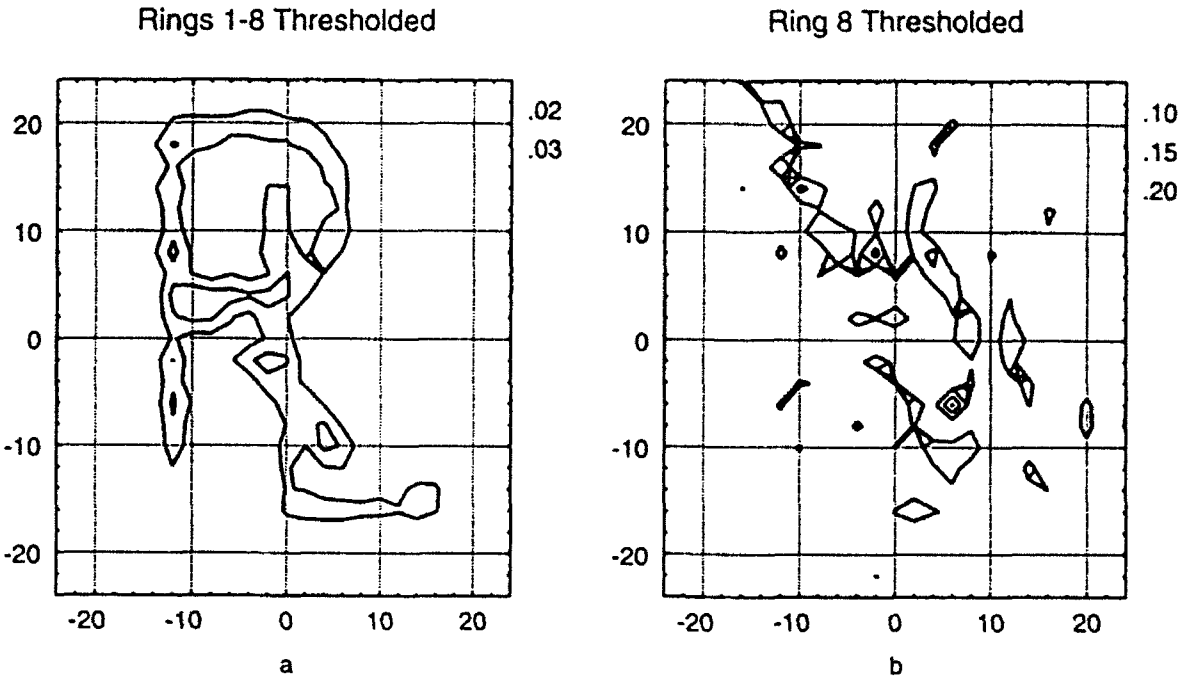


Figure 6: "RL" Target - Conventional Processing, a. Rings 1-8, b. Ring 8.

model. In the latter case, a more detailed electromagnetic based model may need to be incorporated into the imaging process. When noise was added to single ring data, figure 7, both algorithms needed to be thresholded to produce an image resembling figure 6b.

4 Summary

A maximum likelihood based approach to producing images from noisy radar data has been described, implemented on the Rome Laboratory Vax 750 computer, and tested using measured data that was contaminated by simulated Gaussian noise. The original data was measured at the Rome Laboratory Prospect Hill Facility. Results show that with relatively small amounts of data, useful images can be obtained.

Possible future activity could proceed along at least two lines. First, remaining to be analyzed are some data sets from wire-like target structures and

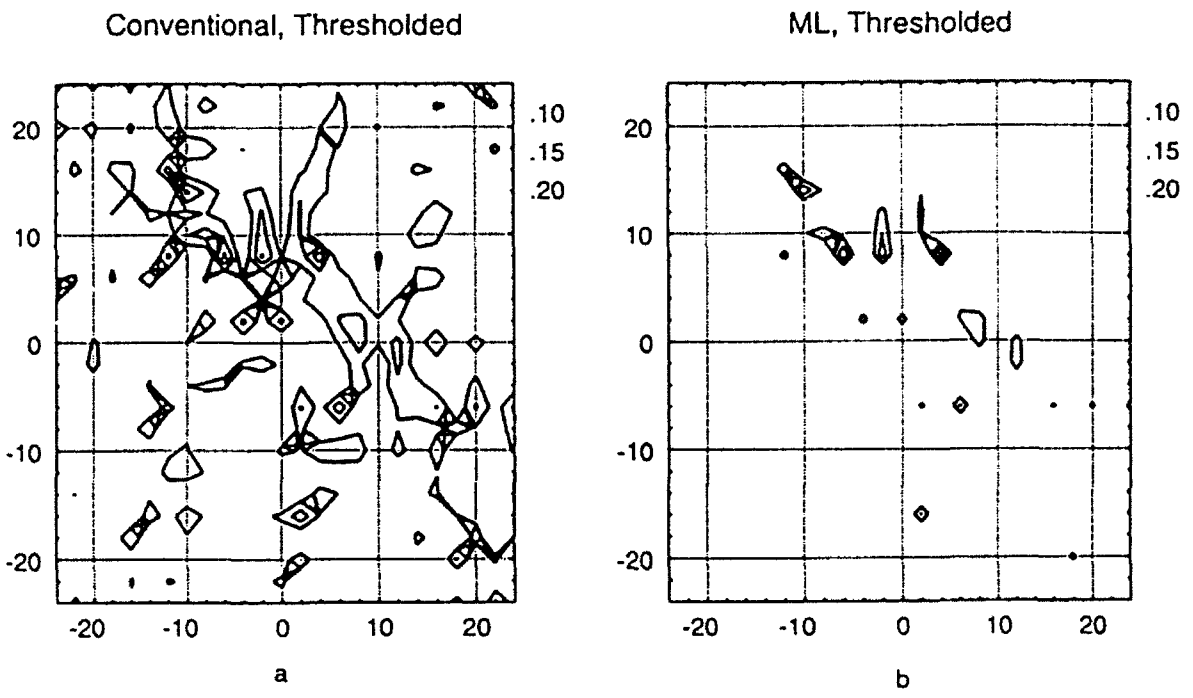


Figure 7: "RL" Target - Ring 8 Image, a. Conventional, b. ML Processing.

from a dimpled "RL" target structure appearing to be of a diffuse nature. It is recalled that the physical process modeled in both types of processing above essentially considers targets as being composed of independent scatterers. It would be valuable to extend the present study to these targets or perhaps others to see how the model works. Another possibility, initially along more theoretical lines, would be to extend the maximum likelihood approach to multiple-rings of data and/or incorporate more detailed scattering mechanisms into the imaging process.

ACKNOWLEDGMENT

The author wishes to thank Robert V. McGahan for making available resouces of the Rome Laboratory Target Characterization Branch to support this work; Dr. Uve H.W. Lammers and Richard A. Marr for providing the measured radar data that was used; and Marc G. Coté for his extensive practical help with the computer and information processing systems.

References

[1] D.L. Mensa, "High Resolution Radar Imaging", Dedham, MA, Artech House, 1984.

[2] W.M. Brown, "Walker Model for Radar Sensing of Rigid Target Fields", IEEE Trans. Aerospace and Elect. Sys., Vol. AES-16, No.1, pp. 104-107, 1980.

[3] U.H.W. Lammers, R.A. Marr, and J.B. Morris, "A Coherent Mechanical Submillimeter Frequency Shifter", Int. J. IR and MM Waves, Vol. 11, No. 3, pp. 367-382, 1990.

[4] U.H.W. Lammers and R.A. Marr, "Narrowband Heterodyne Reception Using Unstabilized Sources", Int. J. IR and MM Waves, Vol. 11, No.6, pp. 701-716, 1990.

[5] U.H.W. Lammers and R.A. Marr, "Doppler Imaging Based on Radar Target Precession", To be published in IEEE Trans. Aerospace and Elect. Sys.

[6] U.H.W. Lammers, R.A. Marr, and J.B. Morris, "Two Dimensional Imaging of Model Targets with a CW Radar", To be published in EASREL International Journal Advances in Remote Sensing.

[7] D.R. Wehner, "High Resolution Radar", Norwood, MA, Artech House, 1987.

[8] H.L.Van Trees, "Detection, Estimation, and Modulation Theory, V.I", New York, Wiley, 1968.

User-Based Requirements for Large-Scale Distributed Information Management Systems: Representation for System Designers

Michael S. Nilan
Associate Professor
School of Information Studies

and

R. David Lankes
Doctoral Student
School of Information Studies

Syracuse University
4-206 Center for Science & Technology
Syracuse, New York 13244-4100

Final Report for:
Summer Research Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D. C.

August 1992

User-Based Requirements for Large-Scale Distributed Information Management Systems: Representation for System Designers

Michael S. Nilan
Associate Professor
School of Information Studies

and

R. David Lankes
Doctoral Student
School of Information Studies

Abstract

This report describes our efforts this summer to generate a method for translating our user-based information system requirements into a representation form that would be readily interpretable by system designers and system analysts. Typically, the kind of requirements specification that we produce from our user requirements analyses are text-based descriptions of problem solving processes as perceived by a group of users. In the past, these text-based descriptions have proven to be difficult for system designers and analysts to interpret. Since our long term research agenda is oriented towards large-scale information management systems which are more complex than traditional applications, we felt that we needed some systematic and easily interpretable format that we could use for communicating our user requirements. Using a combination of hypertext-like representations and a 3-dimensional virtual reality display, we have been able to create a representation system that not only provides for effective interpretation of our user requirements on the part of system designers and analysts, but the virtual reality graphic display configuration also allows us to represent system components (e.g., display devices, databases, network linkages, etc.) in the same graphic environment. We believe that this combination of hypertext descriptions and virtual reality graphic displays will facilitate the accurate representation of user perspectives in large-scale information resource management systems.

SEE VOLUME 4, REPORT NO. 19,
FOR COMPLETE TEXT OF THIS REPORT.

X-BAND T/R MODULE
CONDUCTED INTERFERENCE SIMULATION AND MEASUREMENT
Final Report

Randall H. Pursley
Graduate Research Assistant

Georgia Institute of Technology
Georgia Tech Research Institute
Atlanta, GA 30332

See Summer Faculty Research Final Report 19
Jointly Authored with
Mr. John P. Rohrbaugh

Sponsored by:

Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D.C.

June 1992

X-BAND T/R MODULE
CONDUCTED INTERFERENCE SIMULATION AND MEASUREMENT
Final Report

John P. Rohrbaugh
Senior Research Engineer
and
Randall H. Pursley
Graduate Research Assistant
Georgia Institute of Technology
Georgia Tech Research Institute
Atlanta, GA 30332

Abstract

Conducted electromagnetic (EM) interference measurements and analyses were performed on X-band Transmit/Receive (T/R) modules built by Raytheon and Texas Instruments. The T/R module's Clock, Mode[1], +5 and -7 volt dc supply input lines and the Output Built-in-Test and Evaluation (OBITE) line were evaluated. The Clock and Mode differential input pins are connected within the T/R module to a CMOS gate array through DS8820/7820 differential line receivers. EM interference effects were simulated using PSPICE, and verified through measurements, to determine if the model of a DS8820/7820 provides accurate EM interference simulation results at very high frequencies. The objectives of performing measurements and simulations on the DS8820 were to demonstrate that interference effects can accurately be determined on simpler devices and models prior to developing more complex and costly products, such as T/R modules.

Limited simulations were also performed on the OBITE driver IC (54ALS03 NAND gate) and Power Condition Monitoring (PCM) circuits that are connected to the +5 and -7 volt dc supply lines. The PCM circuits are used to monitor over-voltage conditions on the +5 supply and over-temperature on the transmit power amplifier and to disable receive and transmit modes in the event of over-voltage or over-temperature conditions occur. All interference effects, with the exception of receiver low noise amplifier (LNA) gain compression, could be simulated. Effects that were duplicated during simulation included Mode words not received properly by the T/R module, and the T/R module receiver LNA cycling off and on with the application and removal of interference to the OBITE, +5 and -7 volt supply lines.

Damage effects that were observed while performing the interference mesurements could not be simulated. Two T/R modules and two DS7820 IC's were damaged over the course of this effort.

---

[1] The Data lines were not tested on this effort. The Mode lines are used to send commands to the T/R module to place the module in transmit, receive, etc., mode-of-operation, hence the nomenclature Mode. The Data lines are used to place the T/R module in a particular state-of-operation once the mode-of-operation has been selected. The default state-of-operation was selected, and thus the reason for not testing the Data lines.

Calculating Clock Drift Rates

David L. Sims
Graduate Student
Department of Electrical and Computer Engineering
University of Cincinnati
Cincinnati, Ohio 45221

Abstract

In a collection of physically distinct computers, each one having its own clock, it is often necessary to know how each clock is related to the others and how that relationship changes over time. This relationship is commonly composed of an initial *offset* and a *drift rate*. This summer's project calculated the drift rate between two Encore Multimaxes and two local area network protocol analyzers. Some of the result are graphed that give an approximate drift rate, but no strong conclusions were reached nor was a good drift rate found.

1 Introduction

In a collection of physically distinct computers, each one having its own clock, it is often necessary to know how each clock is related to the others and how that relationship changes over time. This relationship is quantified as follows. At some initial point, two clocks differ by a certain number of seconds. This initial difference is known as the *offset*. Then, after t seconds have elapsed, the two clocks differ by an amount determined by a linear function of t. This function is known as the *drift rate*. These two components, the *offset* and the *drift rate*, are used to quantify the relationship between clocks on physically distinct computers.

This summer's experiment included calculating a drift rate between three pairs of computers. In this experiment, there is one computer, denoted "mmax," that acts as the reference computer. The intent is to derive a drift rate for each of three other machines relative to *mmax*. The three other machines are known as *mach1*, *npac*, and *rome*. These machines will be called the *local* machines while *mmax* will be known as the *global* machine since the drift rates are calculated with respect to a "global" time as seen by the *mmax* machine.

2 Network Configuration

For this experiment a simple linear network is used. Starting at one end of the network, the machine *mach1* is connected to *npac* which is connected to *rome* which is connected to *mmax*.[1]

3 Data Collection

To calculate the three drift rates for the machine pairs *mach1-mmax*, *npac-mmax*, and *rome-mmax*, messages were sent from each local machine to the global machine ("mmax") every hour on the hour. Moreover, each machine recorded the times that other messages passed it by. For example, when *mach1* sent a message M to *mmax*, both *npac* and *rome* recorded the contents of M and the time it passed by.

Several experiments were conducted. Each experiment typically lasted 50 hours or more. During that time, each local machine sent messages to the global every hour on the hour. Additionally, each machine recorded the passing of messages. All the times in these recordings were written with respect to each machine's local clock.

After each experiment, there is one data file residing at each machine that contains all the information that was logged during the experiment. The only exception is the *mach1* machine; it logs no data.

4 Data Alignment

After each experiment three data files have been generated. Each entry in each data file corresponds to exactly one other entry in each of the other two data files. At this point all the data files for all the experiments (three data files per experiment) were collected into a central repository.

A problem with all the collected data was that there was no software to access the data. The next task was to write software to access the data. This software was implemented using C++.

After this softare was written and all the data were fully accessible, it was necessary to write more software to properly "align" the data in the three data files in each experiment. Recall that each entry in a data file corresponded to two other entries. Unfortunately, this correspondence is not concrete. In other words, it is known that a correspondence *exists*, but it is not known exactly which entries in the first data file match up to which entries in the second and third data files.

Given this problem, more software was written to generate a single data file with the corresponding data matched up from the three original data files. The main algorithm implemented by this software is as follows. Every message logged to a data file has a timestamp associated with it. Each timestamp is rounded to the closest hour. Then messages from the three are composed using an algorithm similar to merge sort. The three data files are scanned looking for messages whose timestamps match.

Finally, the three data files generated from each experiment were condensed to a single, coherent data file. It was from this data file that the real calculcations could be performed.

5 Calculations

For each experiment several calculations were made to calculate the drift rates between *napc-mmax* and *rome-mmax*. The graphs at the end of this report plot the calculated drift rates for various experiments on several dates. Graphs entitled "dY Calculation" graph drift rates for *rome-mmax*, and graphs entitled "dX Calculation" graph drift rates for *npac-mmax*.

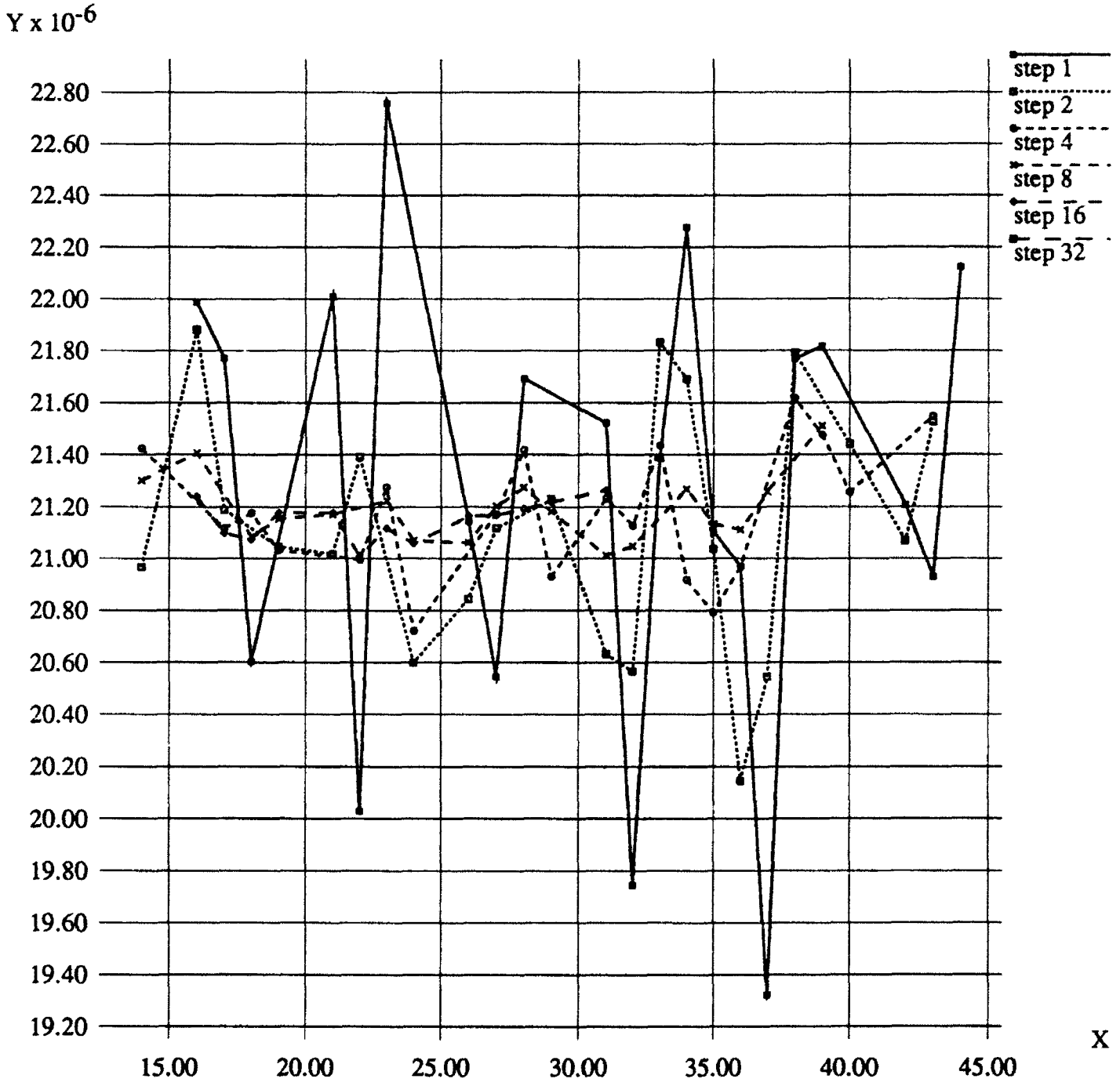
Note that most of the calculated drift rates hover about 20 microseconds per second.

6 Conclusion

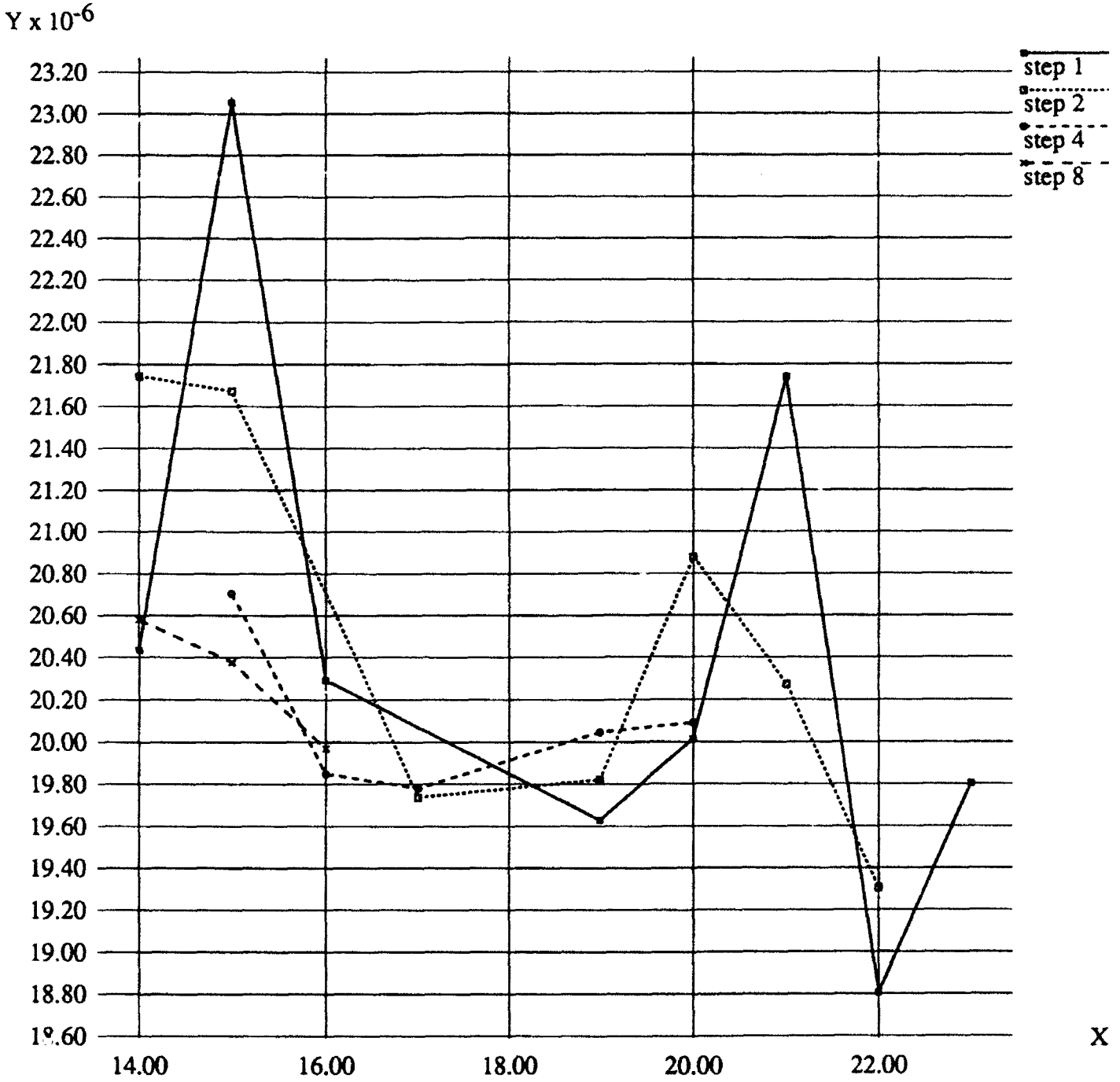
Contrary to the assumption that drift rates would be constant, all the graphs show a different drift rate. The drift rates range from 20 microseconds per seconds to 22 microseconds per second. On one graph the drift rate is actually changing with time.

In conclusion, the resultant drift rates were not what we expected and of little use. Some of the result are graphed that give an approximate drift rate, but no strong conclusions were reached nor was a good drift rate found.

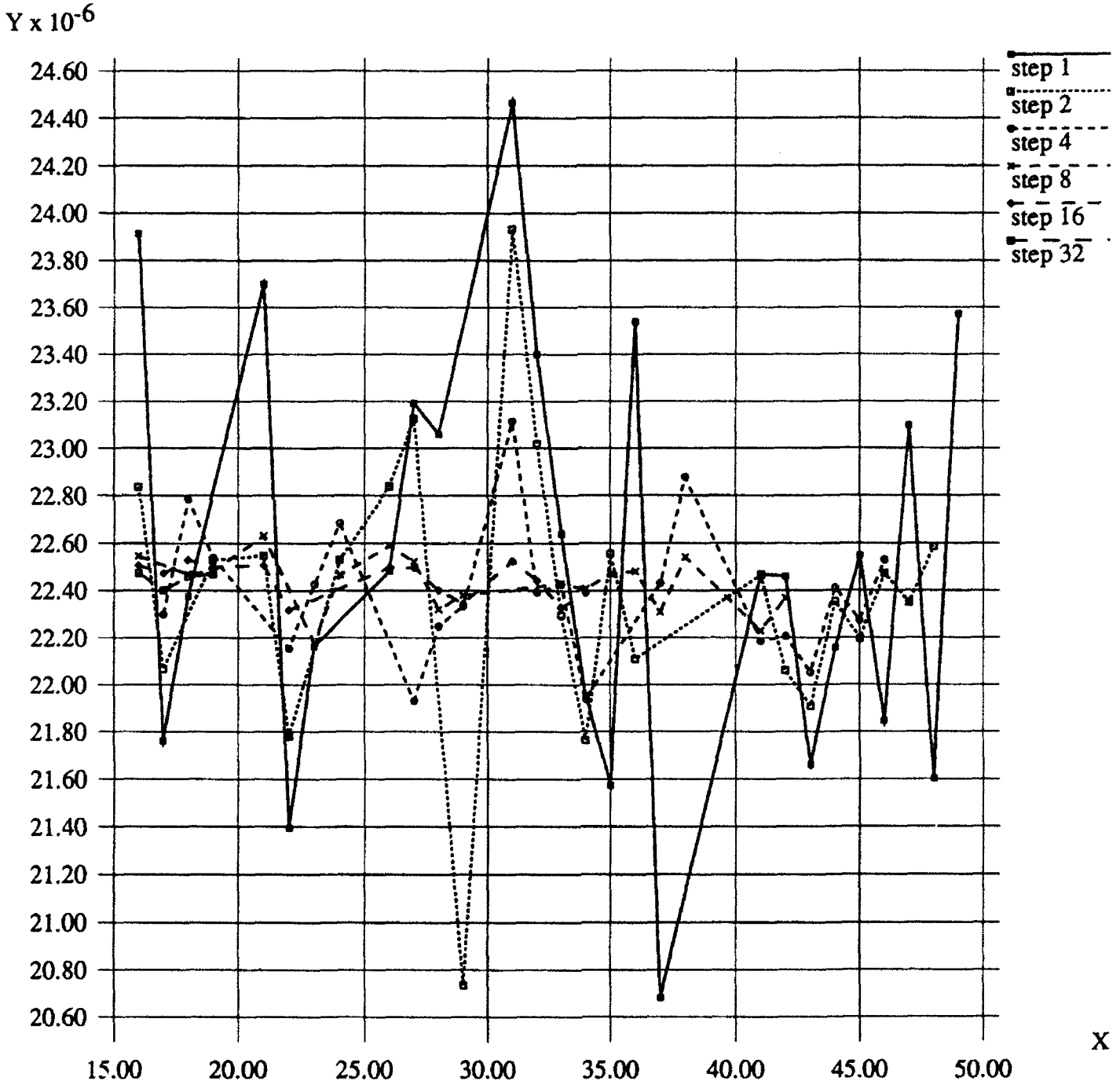
Feb0506, dY Calculation, all steps



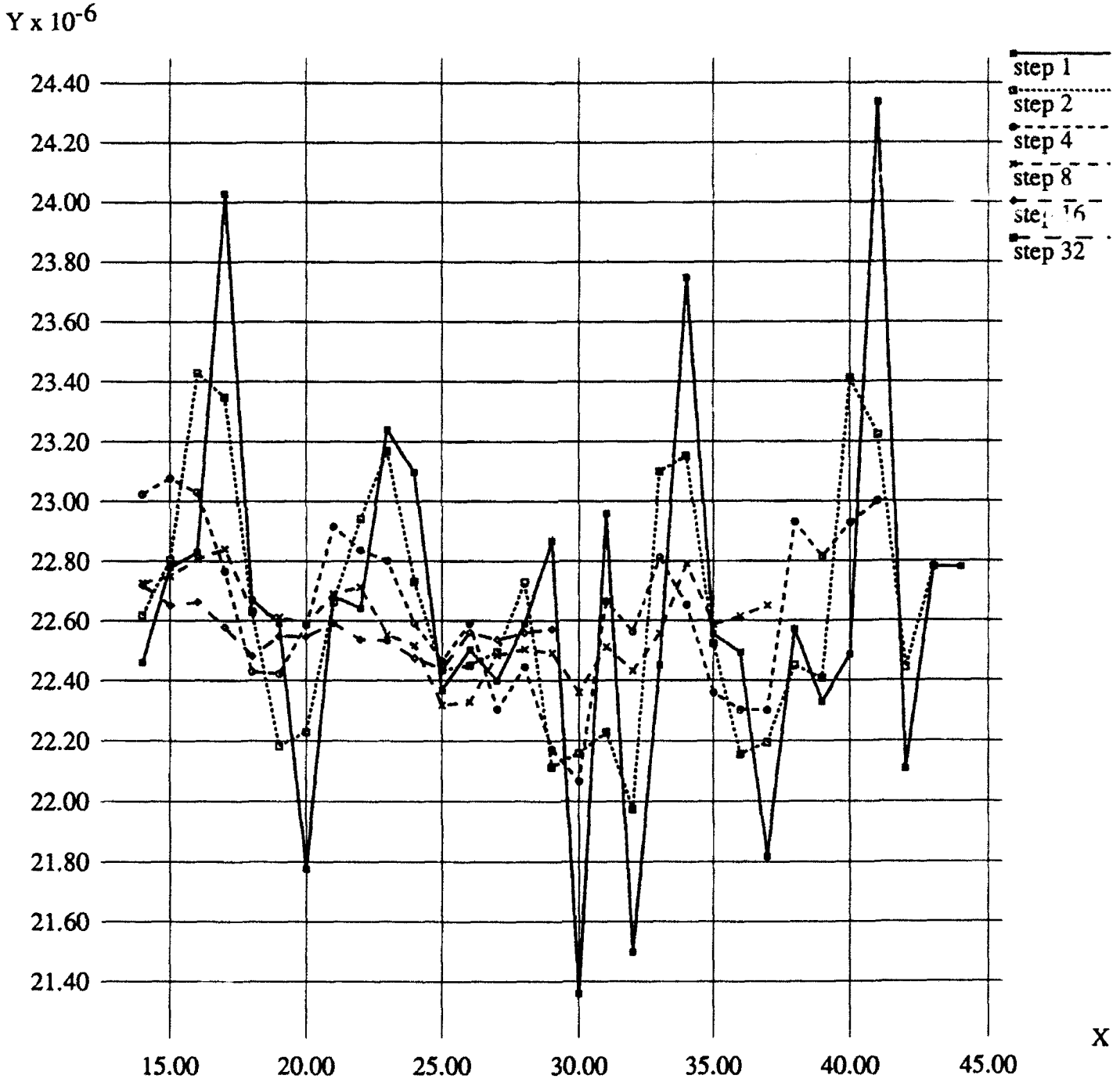
Feb2425, dY Calculation, all steps



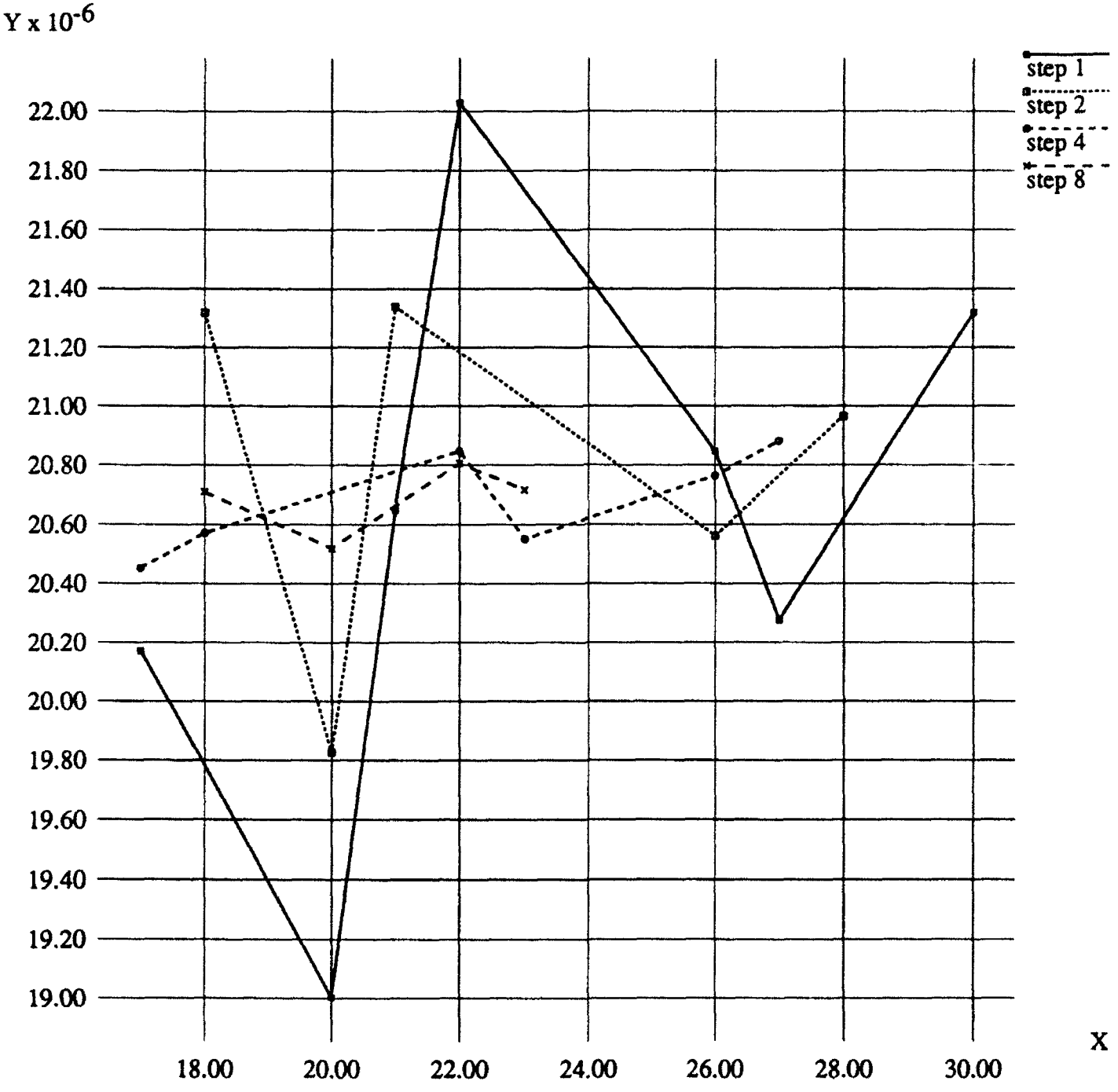
Feb2627, dY Calculation, all steps



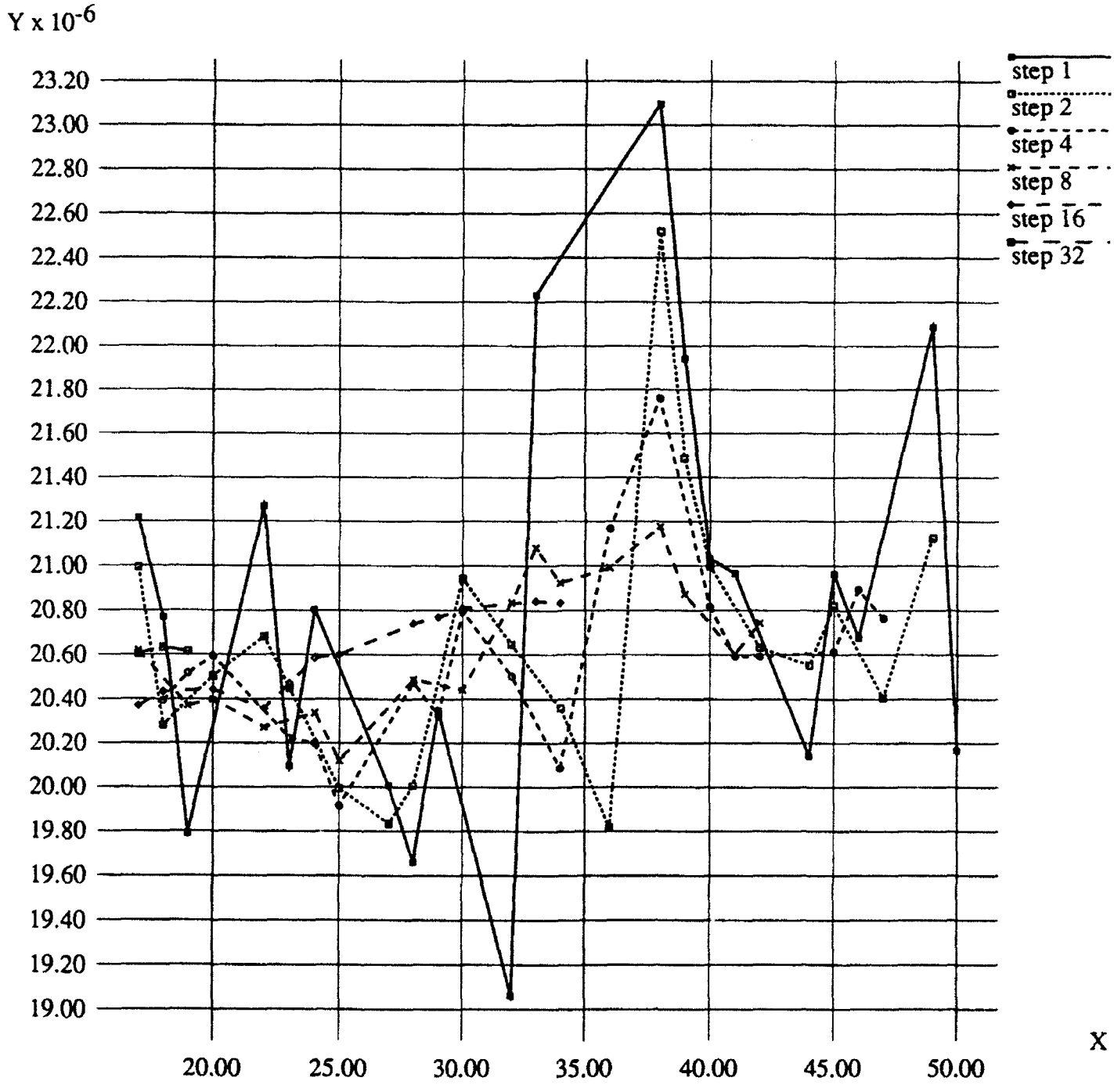
Mar0607, dY Calculation, all steps



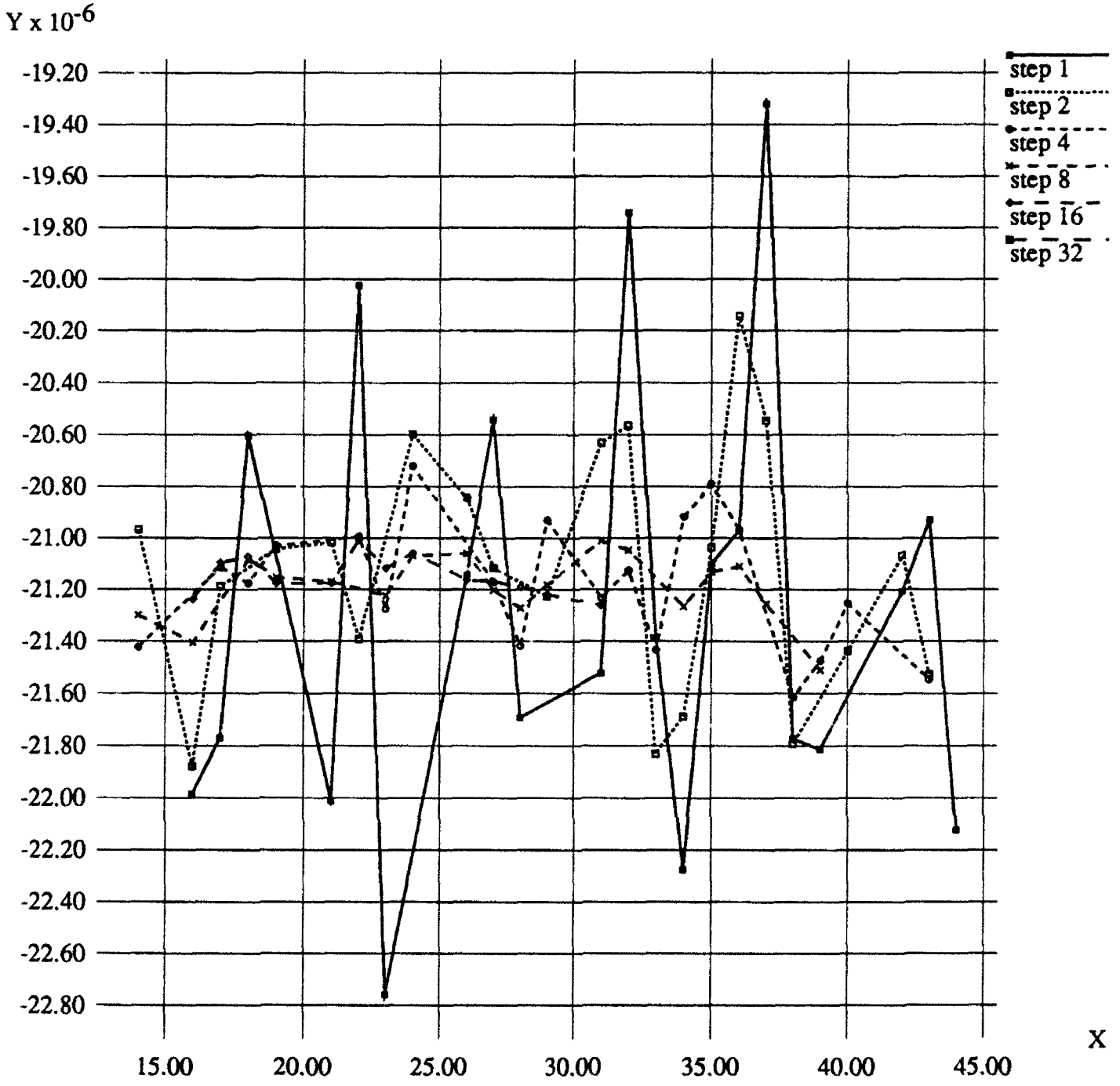
Mar1617, dY Calculation, all steps

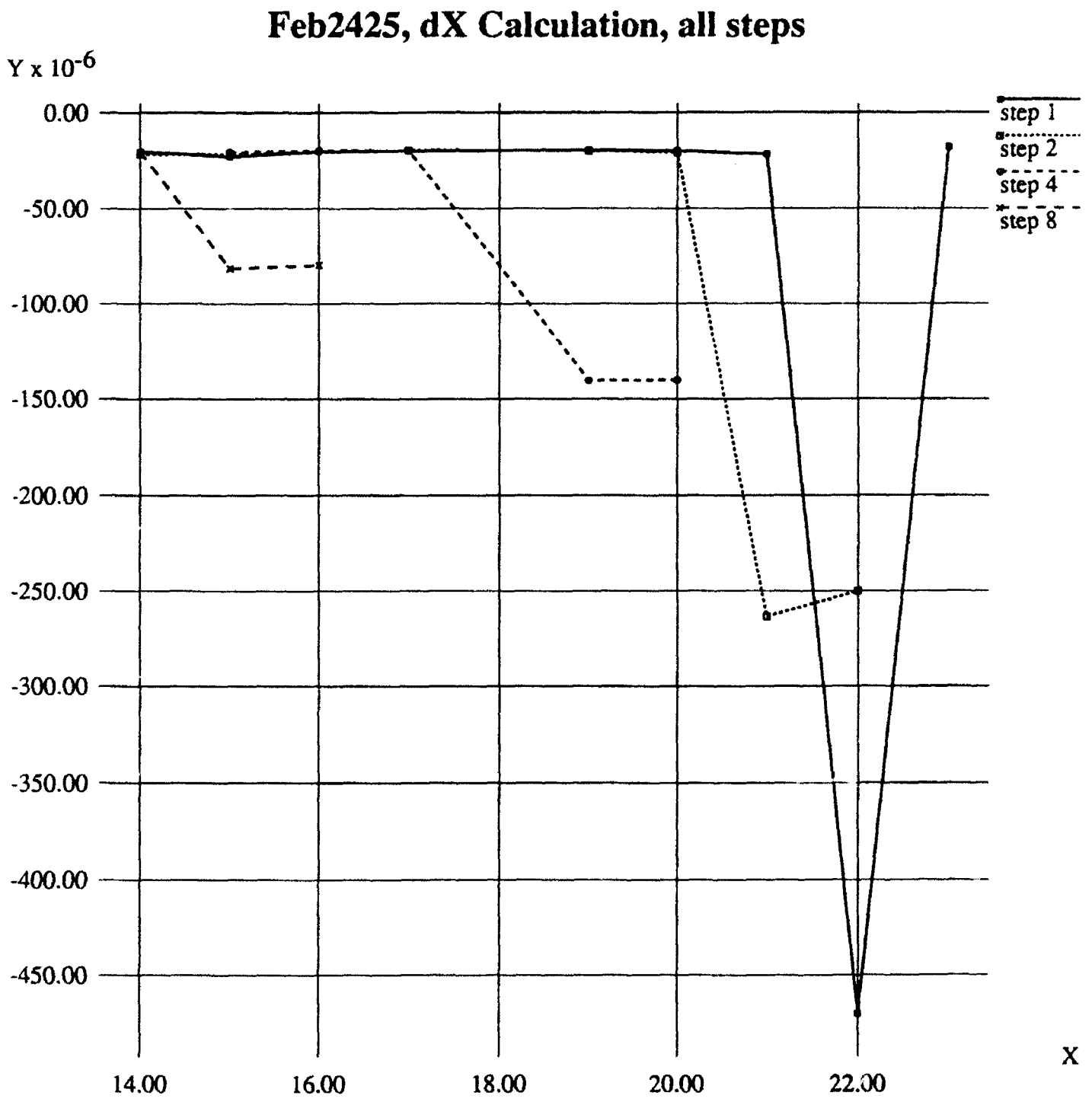


Mar3031, dY Calculation, all steps

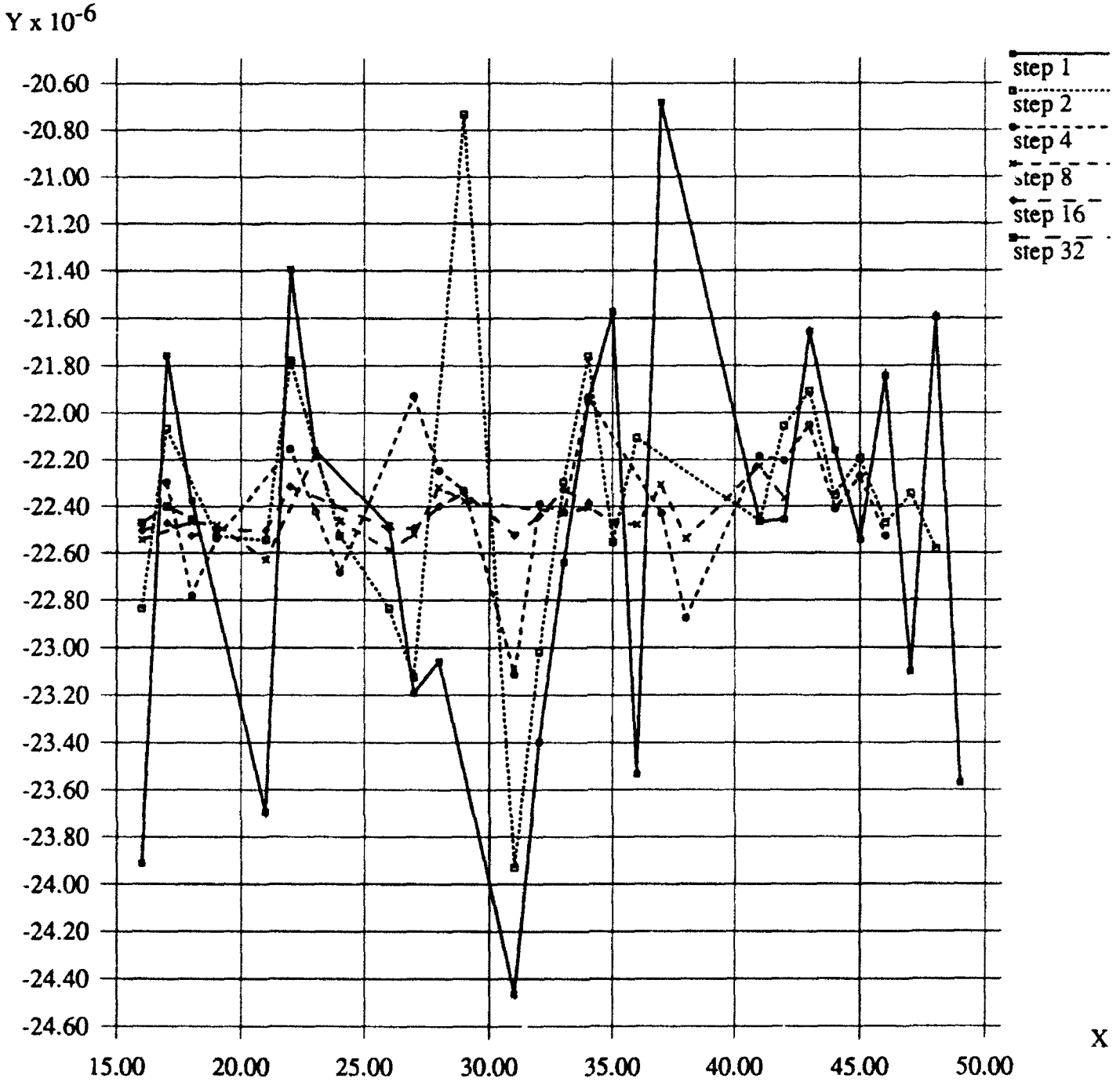


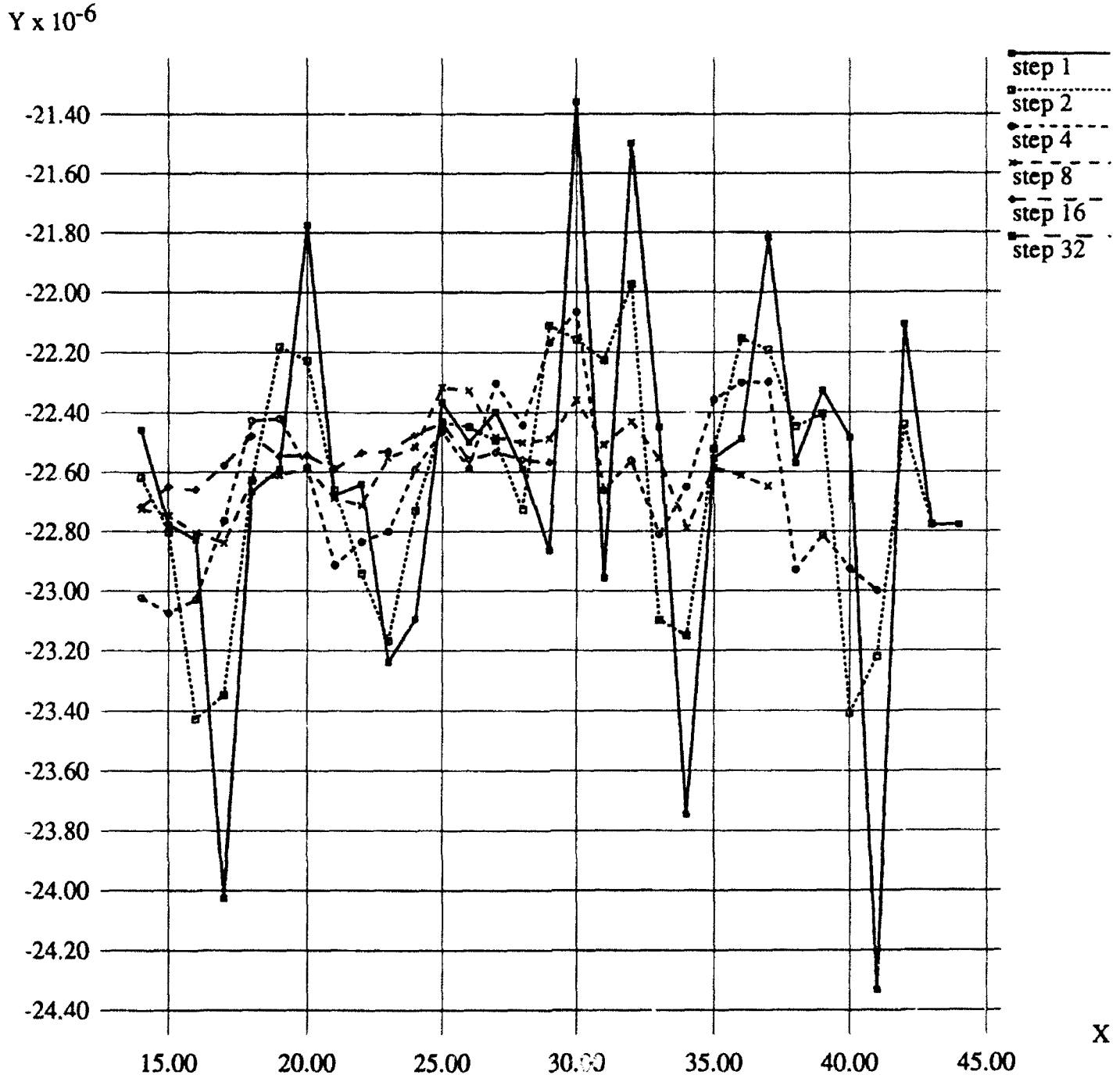
Feb0506, dX Calculation, all steps



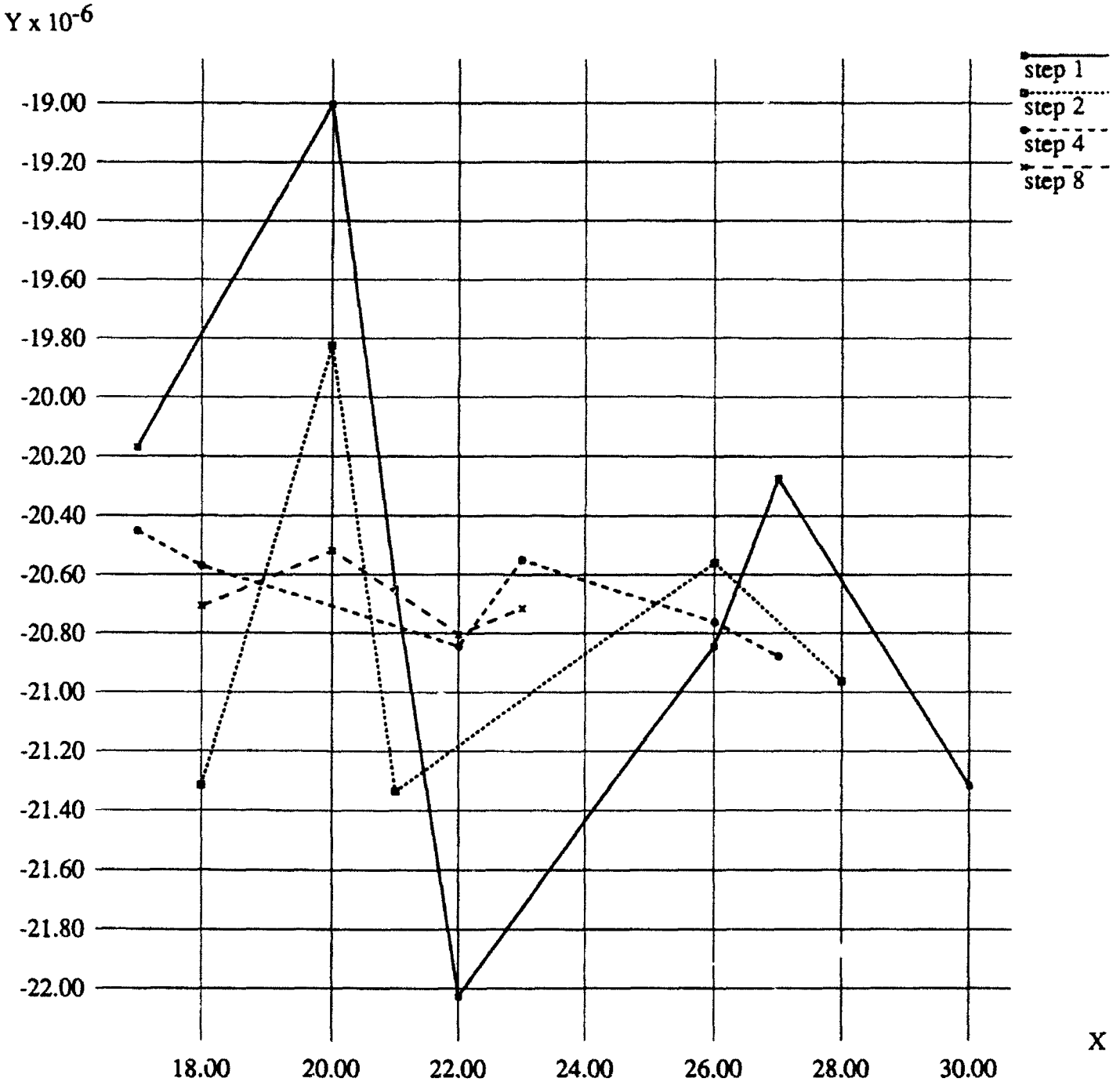

Feb2425, dX Calculation, all steps
Y x 10-6
0.00
-50.00
-100.00
-150.00
-200.00
-250.00
-300.00
-350.00
-400.00
-450.00
14.00
16.00
18.00
20.00
22.00
X
step 1
step 2
step 4
step 8

Feb2627, dX Calculation, all steps

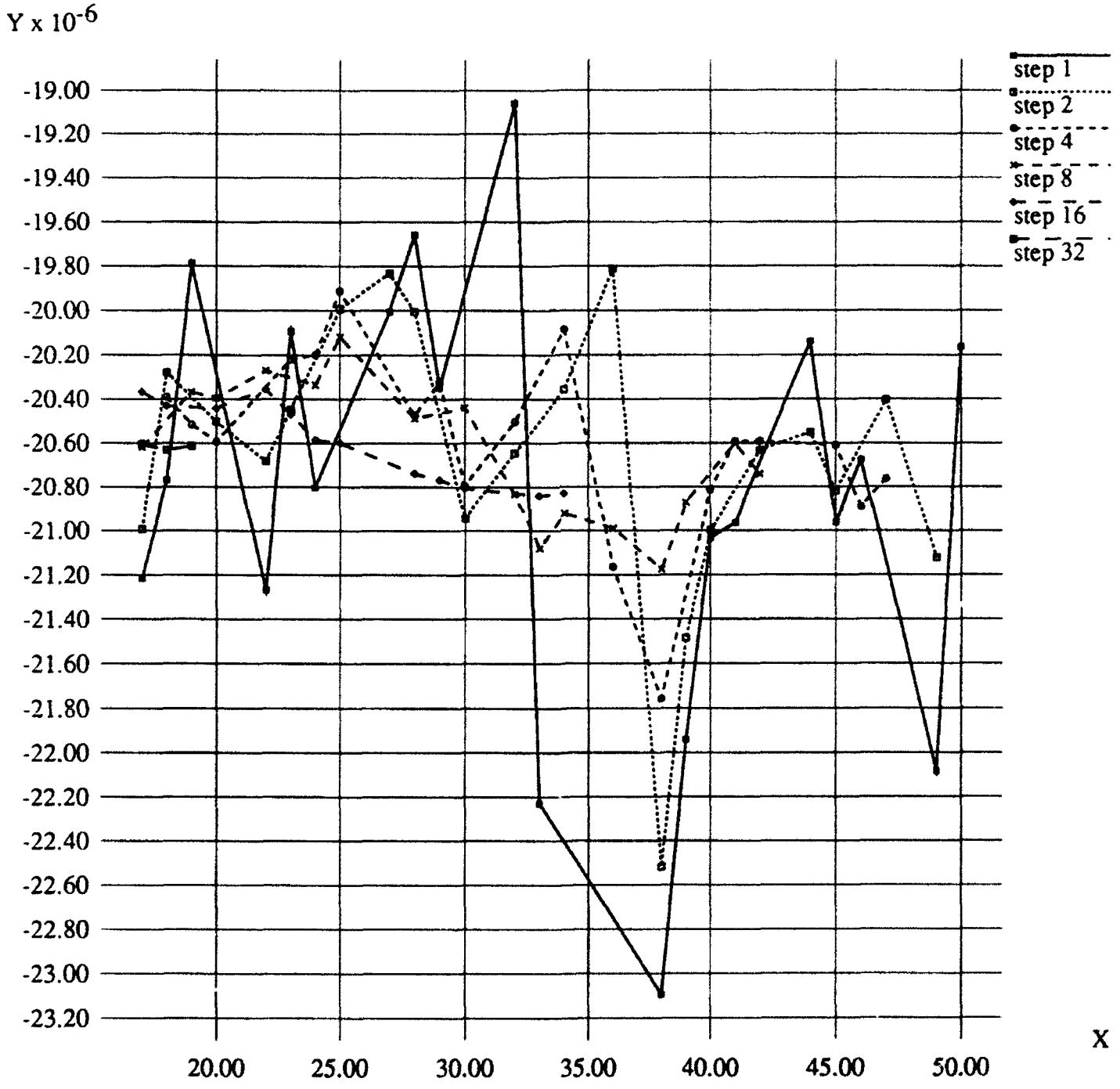



Mar0607, dX Calculation, all steps
Y x 10-6
-21.40
-21.60
-21.80
-22.00
-22.20
-22.40
-22.60
-22.80
-23.00
-23.20
-23.40
-23.60
-23.80
-24.00
-24.20
-24.40
15.00
20.00
25.00
30.00
35.00
40.00
45.00
X
step 1
step 2
step 4
step 8
step 16
step 32

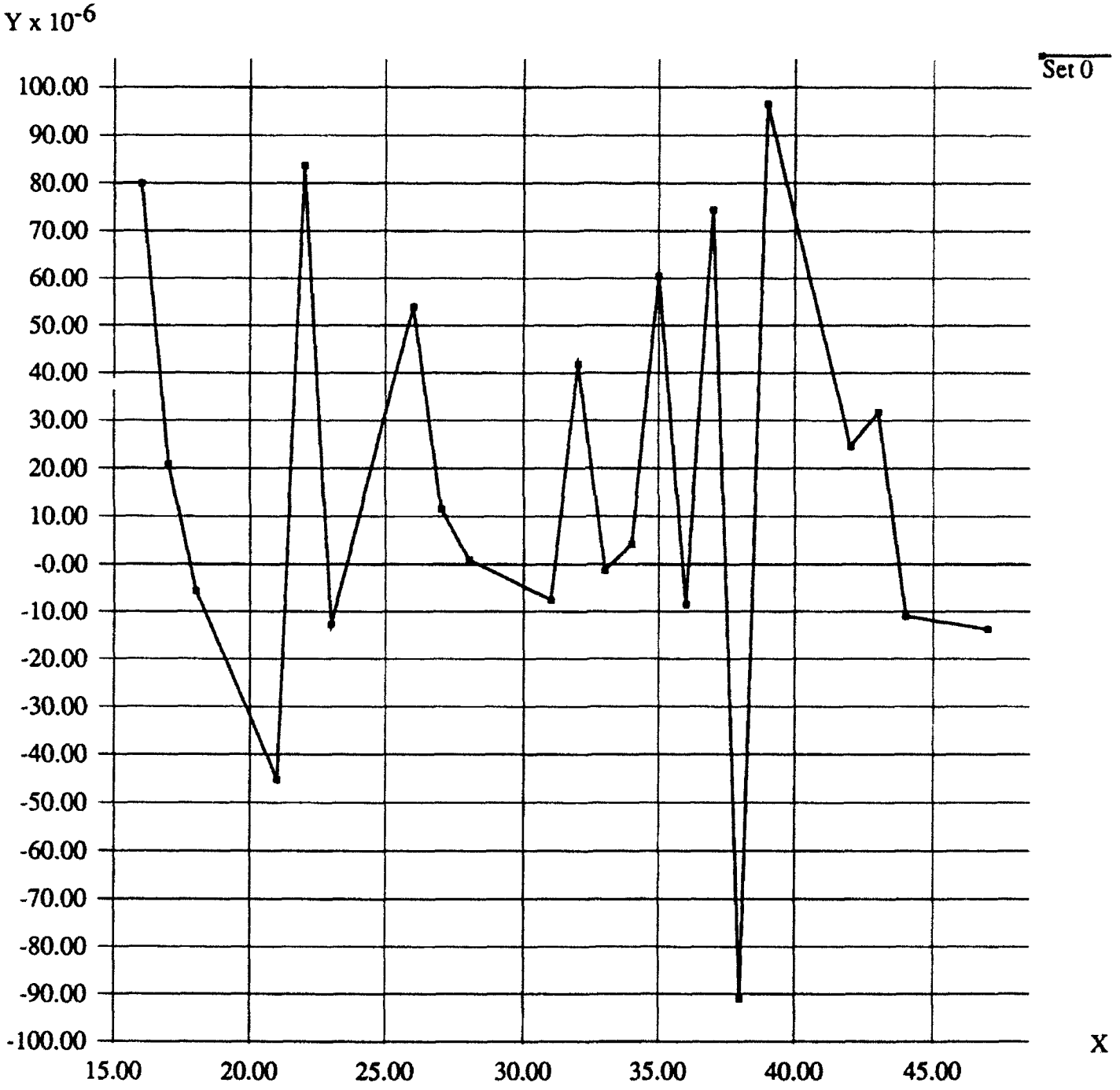
Mar1617, dX Calculation, all steps



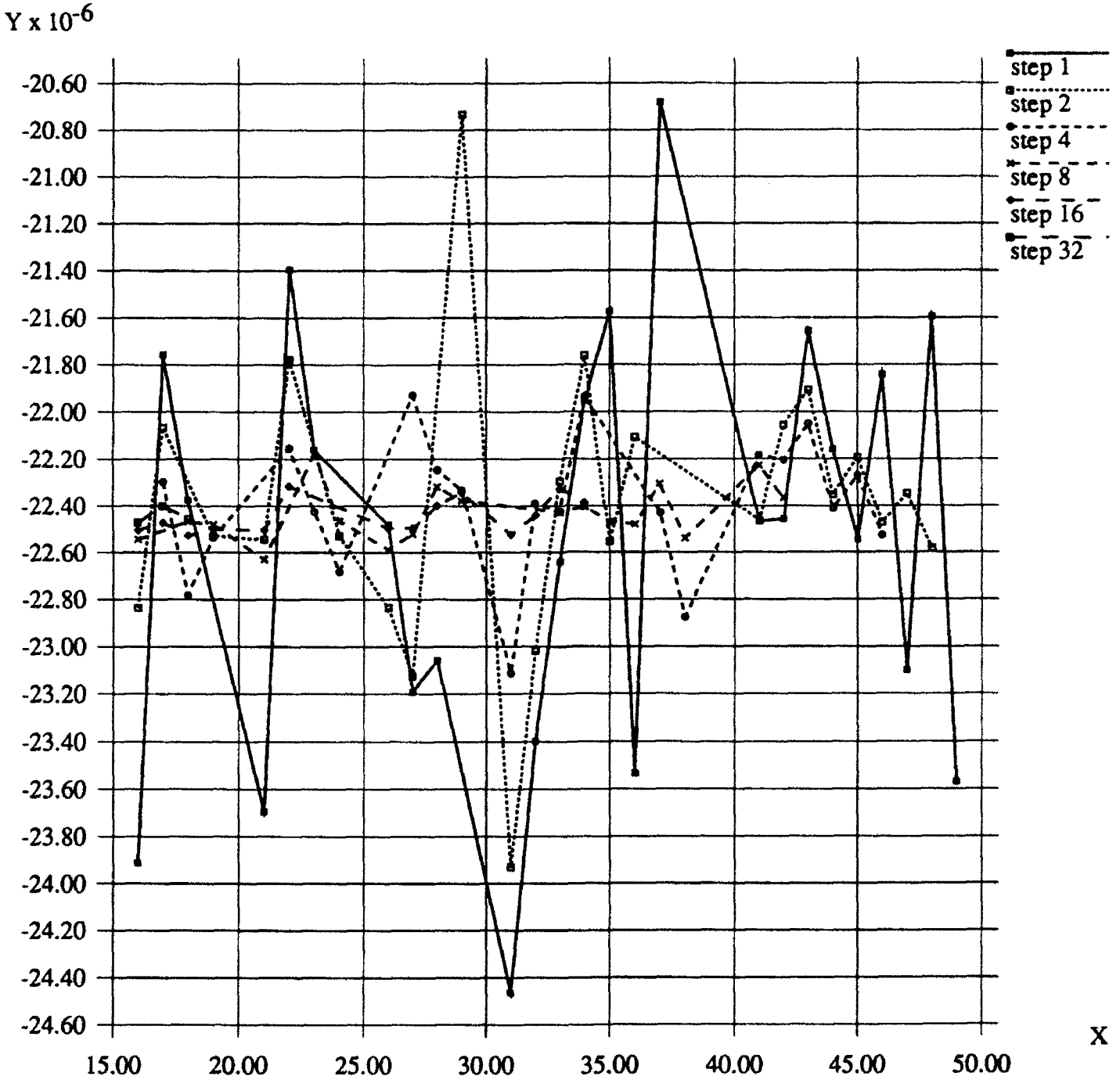
Mar3031, dX Calculation, all steps



Feb0506 (reduced network effects), dX Calculation, step = 1



Feb2627 (reduced network effects), dX Calculation, all steps



CENTRAL ISSUES IN PERFORMANCE EVALUATION OF HETEROGENEOUS DISTRIBUTED COMPUTING SYSTEMS WITH C3 APPLICATIONS

WALEED W. SMARI
Department of Electrical & Computer Engineering

Syracuse University
121 Link Hall
Syracuse, NY 13244-1240

Final Report for:
AFOSR Summer Research Program
Rome Laboratories

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D. C.

September 1992

CENTRAL ISSUES IN PERFORMANCE EVALUATION OF HETEROGENEOUS DISTRIBUTED COMPUTING SYSTEMS WITH C3 APPLICATIONS

WALEED W. SMARI
Department of Electrical & Computer Engineering
Syracuse University
121 Link Hall
Syracuse, NY 13244-1240

ABSTRACT

The principal objective of this research is to contribute to the efficient use of distributed computing systems by identifying some of the major issues that affect the performance of these systems. These major issues are discussed under six categories, namely: system architecture and environment, workload specification and characterization, constraints, service disciplines, performance measures, and finally optimality criteria and strategies. Additionally, this work is aimed at specifying and surveying some key problems and their solutions which we consider crucial to improving the performance - problems such as control, partitioning and mapping, scheduling, synchronization, memory access, among others.

These issues and problems may serve as a crude classification scheme for comparison of distributed computing systems.

CENTRAL ISSUES IN PERFORMANCE EVALUATION OF HETEROGENEOUS DISTRIBUTED COMPUTING SYSTEMS WITH C3 APPLICATIONS

WALEED W. SMARI

I. INTRODUCTION:

A distributed computing system is a collection of processor-memory (hardware) pairs connected by a communications network and logically integrated in varying degrees by a distributed operating system and/or distributed database system. The communications network may be a wide (geographically dispersed) or a local area network.

The widespread use of distributed computing in command, control, and communication (C3) systems is due to many factors. Users are located at different terminals with a desire to communicate and to share information and resources. Information generated at one place is often needed at another. A Distributed Computing System (DCS) **potentially** provides significant advantages some of which are performance enhancement, reliability improvements, availability, resource sharing (both hardware and software resources), scalability, and expandability. This gives the system the ability to easily *adapt* to short-term (varying workloads, failures, network traffics, etc.) as well as long-term (major modifications) changes without significant disruption of the system. Distributed computing systems can provide the necessary power to meet the growing demand of the users community which is growing faster than the advances in devices alone can supply. In the past, a major deterrent to the distributed approach has been cost. However, the cost of hardware is generally going down and cost effective and efficient computer/communications systems are feasible.

These distributed computing systems (along with large distributed databases, distributed processing and distributed communication networks) have given rise to some very complex structures with very complex problems. Let us consider, for instance, the issue of computation speedup. The methods for speeding up the computation may include the following: 1) faster chips - a physics and engineering problem, 2) architectures that permit concurrent processing - a system design problem, 3) compilers for detecting concurrency - a software engineering problem, 4) algorithms for specification of concurrency - a language problem, and 5) models of computation - an analytic problem.

Various distributed computing systems have been proposed, and many have been built. Although technology facilitates the construction of these systems, the way of reaching their full potentials still needs more exploration. It is clear from the above discussion that we need to learn how to think about these systems properly and how to assess and evaluate their performance.

II. FUNDAMENTAL ISSUES AND PROBLEMS

In this section we introduce the primary issues that determine or influence the performance of any distributed computing system. We also discuss key problems and their proposed or implemented solutions – problems such as control, partitioning, scheduling, synchronization, memory access, etc.

A. System Architecture and Environment Issues

This may include:

1. The Number of Machines Available:

The simplest case we can start with is that of a single machine system. For the more practical case of multiple machine systems, we classify them as either parallel machines or shop systems. *Parallel machine systems* assume the situation where each job needs only a single operation for its completion while the machines operate in parallel. If each job requires multiple operations and the machines are used in series, then we refer to these systems as *shops*. In this case, each job needs execution on more than one machine. Three common categories of shop problems can be distinguished:

i. Open Shop Problems: where each job J_j consists of a set of m operations $\{O_{1j}, O_{2j}, ..., O_{mj}\}$. The order in which a job passes through the machines (or the operations are executed) is immaterial.

ii. Flow Shop Problems: where each job J_j consists of a chain of m operations $(O_{1j}, ..., O_{mj})$. Each job in the set of jobs has the same machine ordering.

iii. Job Shop Problems: where each job J_j consists of a chain of m operations $(O_{1j}, ..., O_{mj})$. Each operation has to be processed on a given set of machines. That is, a particular machine ordering is specified for each individual job.

In the more general case of distributed systems, regardless of the point of view (whether hardware or software), a system is actually a combination of both parallel processors and shops.

2. The Types of Machines in the System:

Machine types may be classified as:

i. Identical machines: where all machines have the same speed and capability in processing any job.

ii. Uniform (Heterogeneous) Machines: where each machine has a specific speed of processing regardless of job types. Yet, each machine executes all jobs at the same speed.

iii. Unrelated Machines: where no particular relationship between the processing speeds of different machines (with different jobs) exists. That is, processing speeds are arbitrary.

3. Architecture Configuration:

The architecture as well as the organization of the system greatly influence the kind of

performance modeling and analysis used. Therefore, it is important to define the details of the system's structure and components such as memories, processing units, control, interconnection network, and so on.

Many computer architectures and organizations have been proposed and/or implemented. These may be distributed or centralized systems, pipelined systems, array systems, multiprocessor systems, multicomputer systems, adaptable architectures such as reconfigurable systems or dynamic computer systems, data-flow architectures, etc. Many classifications and taxonomies of computing systems architectures have been published based on the way these systems are viewed. The most celebrated ones were proposed by Flynn [72], Anderson and Jensen [75], Enslow [77], Jones and Schwarz [80], and Duncan [90].

As for components, let us consider for example, the issue of memories. Depending on the organization of the memories, multiple-processor systems can be classified as centralized, distributed, and mixed-memory multiprocessors. In a **centralized-memory** multiprocessor, all memory modules are equally accessible to all processors. A processor-memory interconnection network is needed to allow all processors in the system to access the memory modules. Typical examples are the *Encore's Multimax* and the *CRAY-X/MP*. Several hardware bottlenecks exist in such a system. These are functions of the number of processors, the memory bandwidth, and the bandwidth of the interconnection network used. In a **distributed-memory** multiprocessor, each memory module is physically associated with a processor. No memory is globally accessible. An interconnection network is needed to allow the processors to communicate with each other. Because each memory module is attached to a corresponding individual processor, the performance of an algorithm depends on how well the application problem is *partitioned* and *mapped* into the processors. The main overhead in such a system is the interprocessor communications. A multiprocessor system may have a **mixed-memory** structure to provide a local memory to each processor and global memory modules shared by all processors. An example of such system is the Carnegie-Mellon *Cm** multiprocessor (Gehringer et al [87]).

B. Workload Specification & Characterization

By workload we roughly mean the set of all inputs (programs, data, commands) that a computing facility receives from its users. In any evaluation study, we have to decide under what workload the performance of the system should be evaluated. Performance measures are meaningful only if the workload by which their values are produced is precisely specified.

Two workload-related issues need to be considered during the planning phase of an evaluation study. One is the *specification* of the type of workload which is to drive the system (or a model of the system). The second is the *characterization* of the workload for the system under study. Characterizing a workload for evaluation purposes requires determining which of its numerous

aspects have an influence on the system's performance. An actual workload can be characterized with various degrees of detail using a workload **model**. The workload model is to be formulated, constructed, tested, calibrated, and validated. In a way, this model can be viewed as a set of quantifiable parameters.

Let us discuss next some of the parameters that may be used to characterize a workload.

1. Job Description:

Any actual workload can be regarded as consisting of a set of jobs, each one of which performs an information-processing task when it is processed by the system. Each job that is to be executed on a given system is characterized by certain parameters. These parameters may be *deterministic* or *stochastic*. They may include:

* Number of operations;
* One or more processing times, P_{ij}, that the job J_j has to spend on machine M_i on which it requires processing;
* A due date or deadline, $d_j > 0$, by which the job J_j should ideally be completed;
* A release (or ready or arrival) date, $r_j \geq 0$, on which the job J_j becomes available for processing;
* A weight, $w_j > 0$, indicating the relative importance of the job J_j;
* A completion time, $C_j > 0$, by which the job J_j is actually completed;
* A nondecreasing real cost function to measure the cost incurred if the job is completed at time t.
* An allowance time, a_j, which is the period allowed for processing between the ready time and the due date: $a_j = d_j - r_j$.

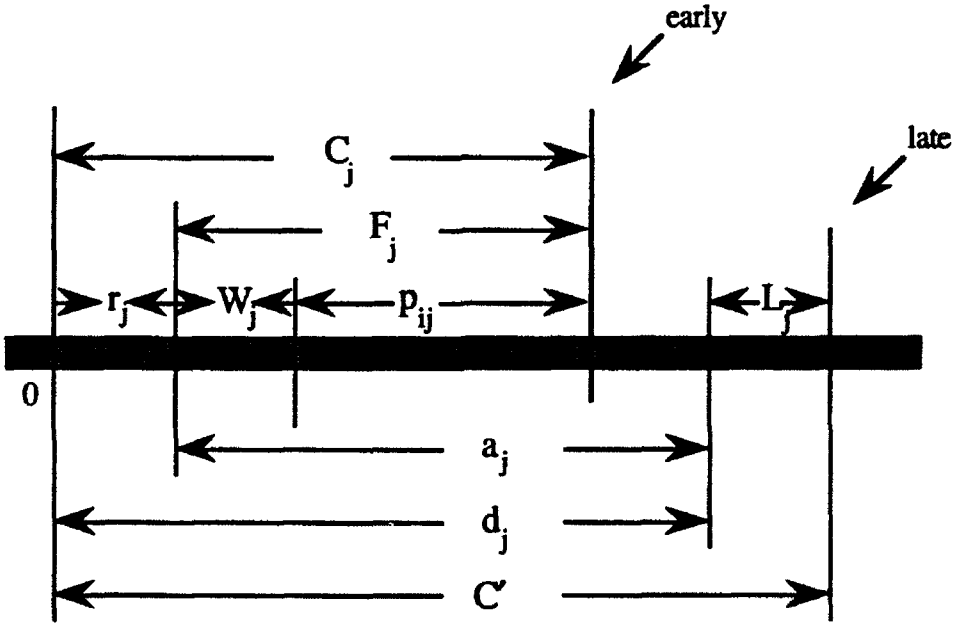


Fig. 1 Time Quantities of a Job

Various other terms associated with a job J_j may be defined as following:

Flowtime: $F_j = C_j - r_j$

Waitingtime: $W_j = F_j - P_{ij}$

Lateness: $L_j = C_j - d_j = F_j - a_j = C_j - r_j - a_j$

Earliness: $E_j = \text{Max}\,[0, d_j - C_j] = \text{Max}\{0, -L_j\}$

Tardiness $T_j = \text{Max}\,[0, C_j - d_j] = \text{Max}\{0, L_j\}$

$$\text{Tardiness indicator} = \begin{cases} 1 \text{ if } T_j > 0 \\ 0 \text{ if } T_j = 0 \end{cases}$$

2. Job Behavior:

As was mentioned before, the nature and structure of the workload affect to a great deal the performance of the computing system. For example, in some computing facilities, as in hard real-time systems, there are two types of tasks: nonperiodic and periodic tasks. A **nonperiodic** task has arbitrary arrival time. A **periodic** task with period P requires that one instance of the task would be executed once every P units of time after system initialization. The deadline for each request of a task can be no later than the initiation of the next request of the same task, i.e. no later than P. Such is the case in a satellite tracking system where, in order to have the antenna continuously aimed at a satellite, the system must process periodic requests for adjusting the aiming of the antenna. Each request for adjustment must be completely processed by the tracking system within a limited time with respect to the previous adjustment of the antenna. In this case, the tasks are *periodic* and with *hard-real-time* requirements.

Jobs may be in various states of processing at any given time. The following terms account for this:

$N_w(t)$ = the number of jobs waiting or not ready for processing at time t.

$N_p(t)$ = the number of jobs actually being processed at time t.

$N_c(t)$ = the number of jobs completed by time t.

$N_u(t)$ = the number of jobs still to be completed by time t.

Therefore: $N_u(0) = n$, $N_u(C_{max}) = 0$, and at any time t,

$$N_w(t) + N_p(t) + N_c(t) = n$$

$$N_w(t) + N_p(t) = N_u(t)$$

Generally speaking, when using a distributed computing system, the workload of the system differs in a variety of ways from the centralized system. However, very little data is available to conduct a quantitative analysis of the workload of these systems.

C. Types of Constraints

In any computing system, one or more constraints may be present. The two principal categories are: constraints *in the system* and constraints *in the tasks* to be executed. These two may additionally be classified as constraints in time, priority, precedence, resources, and interprocessor communication.

1. Time Constraints:

A time constraint affects the execution timing of an operation. A typical such constraint may be a *due date* which imposes restrictions on the latest allowable completion time. Another example is a *release time* constraint which specifies the earliest starting time for processing of the job.

Timing constraints are especially important in a real-time system. Such a system violates a timing constraint if any job misses a deadline. Depending on how much it tolerates violations of timing constraints, a real-time system can be classified as: "hard" or "soft". A **hard real-time** system cannot tolerate a single violation of a timing constraint. A task is considered to be of value only if it finishes before its deadline. In contrast, a **soft real-time** system tolerates timing violations and allows jobs to be prioritized according to their contribution to the continuing function of the system while minimizing the total number of deadline misses. Performance, in this case, is characterized by the extent to which jobs miss deadlines. In effect, jobs have to be executed as quickly as possible, but there is no explicit timing constraint associated with them.

2. Priorities:

In a priority-driven environment, the active task with the highest priority will be processed first. An active task is a task that has requested processing but has not yet received an amount of processor time equal to its run-time. A *priority-driven algorithm* is characterized by the manner it assigns priorities to tasks. The assignment can be carried out either statically or dynamically. If fixed priorities are assigned to tasks once and for all, then the priority assignment is said to be **static**. A fixed priority list of jobs can be constructed by sorting the tasks in decreasing order of their priorities. At any time t, when a request for processing is initiated or when the processing of a task is completed, the priority list will be scanned from the top to find the first active task in it. Once it is located, this highest priority active task will be processed next.

On the other hand, if priorities of tasks can change from request to request, then the priority assignment is said to be **dynamic**. The implementation of a dynamic priority-driven algorithm is similar to that of a static one except for the use of a *dynamic priority list*. At any time t, the first task in the list, J_j, will be processed. When J_j is completed, it will be removed and the new first task in the list will be processed. If during the processing of J_j a new request for a task J_k is initiated, then task J_k will be inserted into the list by comparing its priority with the priorities of the tasks that are already in the list. If J_k is the new first task in the list, then it will be processed immediately when

preemption is allowed; otherwise, the processing of J_j will continue.

A typical priority-driven algorithm is the earliest-deadline-first algorithm.

3. Resource Constraints:

A multiprocessor system has various types of resources, say r types. These resources may be different kinds of processors, memory modules, buses, and so on, with distinct characteristics. We specify the resource requirements of job J_i, by a function $R(J_i)= (R_{i1}, R_{i2}, ..., R_{ir})$, with R_{iy} being the number of units of the y-th resource that is needed by task J_i during its execution. These resources in $R(J_i)$ are assumed to be used simultaneously during the execution of job J_i.

If we denote the number of units of the y-th resource type in the system by $R_y > 0$, $1 \leq y \leq r$, with $R_y = \sum_i R_{iy}$, $1 \leq i \leq n$, then the total resource availability in the system will be specified by the vector $\mathbf{R}= \{R_1, R_2, ..., R_r\}$. The existence of resource constraints require that the total number of resources of the various types which are needed by the jobs being processed at any given instant of time, do not exceed the total available amount of resources as specified in the vector **R**.

4. Precedence Constraints:

In many situations it is assumed that tasks are independent in the sense that the processing of a task will not depend on the processing of other task(s). This means that all necessary information and data required for the processing of the tasks are self-contained. The independence assumption greatly reduces the complexity of the problem.

However, in many applications, restrictions on the order in which tasks can be executed arise naturally. Consider two tasks J_i and J_j with the property that the execution of J_j will require some information from the execution of J_i. This means that J_i must first be executed to generate the necessary data for J_j. Consequently, each processing of J_j must be preceded by a complete execution of J_i. Normally, this kind of precedence relation between two tasks extends to a larger set of tasks.

The precedence relation is a nonempty *partial ordering* relation defined on the set of tasks. The partial ordering relation is modeled by a *directed acyclic graph* (**DAG**), G. Each task is represented by a vertex of G. A directed edge from J_i to J_j indicates that J_i must be processed before J_j. Job J_i is said to be a *predecessor* of J_j and J_j is said to be a *successor* of J_i. Precedence constraints can also be expressed as an ordered pair $(J_i, J_j) \in R$, where R is a precedence relation which lists all allowable orderings between jobs J_i and J_j.

In scheduling a given task, the precedence constraints induced by the partial ordering relation on the set of tasks must be satisfied. If a predecessor of job J_i has not been completely executed, then the processing of J_i must be delayed and the task is said to be **blocked**. When all the predecessors of J_i have been completely executed, the processing of J_i can be started as soon as scheduled, and J_i is said to be **unblocked**. Thus, at any instant, task J_i can be classified as being in one of the following four states:

i. *Inactive state*: the current request for J_i has already been completed and J_i is waiting for its next request to be initiated;

ii. *Ready state*: a request for J_i has been initiated and the task is waiting to be unblocked;

iii. *Active state*: a request for J_i has been initiated and the task is unblocked;

iv. *Executing state*: task J_i is currently being processed.

A precedence relation may be tree-like, forest-like, or arbitrary: any non-cyclic precedence relation.

5. Synchronization & Communication Issues:

From the viewpoint of processes, there are two basic process synchronization and communication models:

i. The *shared-memory* model in which the system has a global memory accessible by all processors. The processes communicate through shared variables. In such a system, the access time to a unit of data is the same for all processors. A hardware device or a software protocol is required in such systems for arbitrating the access to the memory among the processes sharing it. The shared-memory may cause a software bottleneck.

ii. The *message-passing* model in which processors communicate by passing messages explicitly through an interprocessor communication network. The performance of an algorithm in these systems depends on how well the application problem is **partitioned** and **mapped** into the processors, and on the **efficiency** of the communication mechanism. A distributed-memory multiprocessor system does not have the software bottleneck as in the case of a shared-memory system; however, it does experience interprocessor communication problems.

The interprocessor communication mechanism and the computational power of the individual processor are two of the major factors which affect the performance of the system. In order to fully explore the power of a distributed-memory multiprocessor system, there must be a balance between computation and communication. Shih and Fier [87] suggest that an ideal ratio of 1:1 between computation and communication should be achieved. It has been shown though (Dunnigan [87], Heath [87]), that the communication/computation ratios of some of the present distributed-memory systems are very high (> 10). This indicates that the communication mechanism provided in these machines does not match the speed of powerful processors and thus becomes the major bottleneck of the system performance. This problem becomes more critical as processors continue to become faster and more powerful. Therefore, in order to improve the overall system performance, the communication overhead must be significantly reduced, and efficient methods of handling all types of communication should be introduced.

Information items communicating in multiprocessors include synchronization primitives, status semaphores, interrupt signals, variable-size messages, shared variables, and data values.

Three basic communication strategies are in use:

(i) *Unicast (One-to-One)* communication is the sending of a message from a source node to one destination node. This type of communication is directly supported by all distributed-memory multiprocessors.

(ii) *Broadcast (One-to-all)* communication is a type of information exchange in which a source node wishes to send a message to all other nodes in the system (Ho & Johnson [86]).

(iii) *Multicast (One-to-Many)* communication where a node wants to send the same message to k other nodes.

To send messages to the right destination at the right time is very important in many applications, since a node may have to await a message from some other nodes before continuing its computation. Therefore, it is desirable that:

(a) each individual destination receives the message through a shortest path; and

(b) the number of intermediate nodes required to deliver the source message to all destinations is as small as possible, in order to reduce the traffic created by the multicast communication in the network (Lan et al [88a, 88b]).

When a message delivery between non-neighboring nodes occurs, the message has to be forwarded through some intermediate nodes. Two transport mechanisms are currently used:

(1) Circuit switching: where a physical communication path between the source and the destination has to be established first. Then, the source can send out the message to the destination. The routing overhead is paid only once at the circuit set-up time. Also, no link along the established path can be shared by another message delivery.

(2) Packet switching: No physical path is established before the starting of a communication. A message is decomposed into packets which are sent out individually. The source determines its output link(s) and sends the message to the neighboring node(s). Then, each of the nodes which receives the message will decide its output link(s) for further forwarding, and so on. One link is requested at a time, and released immediately after it is used. Routing decisions need to be made for each packet. A buffer is needed at each node to temporarily store the message. The efficiency of the communication depends on the strategy for making the routing decisions as well as the size of the packets.

When message passing is used as the means for interprocessor communication, if a node wants to send a message to a neighboring node, the message delivery is relatively simple. However, if a node wants to send a message to a distant node, the message has to traverse through some intermediate nodes. To send a message from a node to several other nodes, the situation becomes more complicated. One of the major problems in interprocessor communication is to determine which path(s) should be used to deliver a message from a source node to some destination node(s), i.e. message routing. **Message routing** techniques may be classified as centralized or distributed. In *centralized routing*, the source node determines all the intermediate nodes for message delivery. The

addresses of all intermediate nodes must be tagged onto the message, and hence a particular path is specified for each of the destinations. This will create extra communication overhead, especially in the case of multi-destination message routing. In *distributed routing* the source node and each node involved in message forwarding specifies only which of its neighboring node(s) are to be involved in the message delivery.

D. Service Disciplines

1. General Overview

By service disciplines we mean the order in which the arriving tasks are handled at a processing unit.

There are two basic schemes. If the service discipline allows jobs to be interrupted by new arrivals, we speak of a service discipline allowing **preemption.** If such an interruption is not allowed, the service discipline is referred to as **non-preemptive.** In the case of preemption, if processing of the interrupted job can resume where it left off without loss of information, it is a *preemptive-resume* service discipline; if the job has to be started again, we speak of a *preemptive-repeat* service discipline.

Numerous service policies are discussed in the literature. The most common and simplest service discipline is the *First-Come-First-Served* (**FCFS**), where the jobs are handled in their order of arrival. It is a policy that favors the longest waiting job irrespective of the amount of service time demanded by it. Average turnaround time and average waiting time of jobs are generally not minimal, especially for short jobs. The policy is also unfair in that unimportant jobs make important jobs wait. Another common service discipline is the *Shortest-Job-First* (**SJF**). Here, associated with each job is the length of its burst (estimated service time). The job with the shortest burst is the next one to receive service. This policy minimizes the average waiting time of a task. The main difficulty with SJF is that it can effectively prevent jobs that require long time from receiving service. Preemptive SJF is called *Shortest-Remaining-Time First* (**SRT**). SRT will preempt the currently executing job and start executing a new job just arrived with a shorter burst than what is left of the currently executing one. An algorithm which is a balance between FCFS and SJF extremes is the *Highest Waiting Ratio Next* (**HWRN**). It gives fairly quick response to the short jobs, but also limits the waiting time of longer jobs (aging). The service discipline which handles its jobs in the reversed order of arrival is referred to as the *Last-Come-First-Served* (**LCFS**). LCFS can be applied to preemptive scheduling, so that the jobs in service are interrupted by each new arrival.

If there are groups of jobs at a processing unit, and one group has priority in accessing the servers over another group, then we have a **priority** service discipline. The servers are allocated to the tasks with the highest priority. SJF can be regarded as a special case of priority scheduling. Priorities can be defined externally (for example: the kind of account the user has) or internally (such as time limits

or memory requirements $\cdots$ etc.).

Another common scheduling algorithm, which is preemptive, is *Round Robin* (**RR**). Here, the processor's time is equally divided over all jobs in the queue. This is done by dividing the processor's service into quantums of time with each quantum = 1/N units, where N is the number of jobs in the queue. When a job has used up its time-share, it will be preempted and added to the end of the ready queue, while the next job is placed at the beginning of the queue. This exchange of jobs is called **swapping**, which is the main source of overhead in RR and in preemptive policies in general. In its extreme cases: if the quantum is $\rightarrow \infty$, then we have the FCFS discipline; if on the other hand the quantum is $\rightarrow 0^+$ (e.g. executes only one instruction) then we have *processor sharing* (**PS**). Processor sharing is not practically feasible; nonetheless, it is used in system analysis to approximate complex situations under specific assumptions. Essentially, PS behaves as if each of the n jobs has its own processor running at 1/n the speed of the real processor.

There are other scheduling policies used in such cases as bus arbitration or page replacement in virtual storage management. For instance, the *Least-Recently-Used* (**LRU**) strategy gives the highest priority to the requesting processor that has not used the bus for the longest time. This assumes that past behavior is a good indicator of the future. LRU requires a substantial overhead and hardware assistance since priorities must be reassigned after each bus cycle. Strategies that approximate LRU are available which have lower overhead and hardware requirements.

Another class of service disciplines has been introduced for situations in which jobs are easily classified into different groups based on some criterion such as different response time requirements. A **multi-queue** scheme partitions the ready queue into separate queues (System tasks, Interactive, batch, and so on), and each job is assigned once and for all to one of the queues based on some property of the job. Each queue has its own scheduling policy. In addition, there is a scheduling policy between the queues.

In **multi-level feedback queues**, each job is allowed to move between queues. A new process initially enters at the back of the top queue. It then moves through the queue according to the queue service discipline , until it gets served. If the job is completed, or it is waiting for I/O completion, then the job leaves the queue. If the quantum expires before the job finishes execution, the job will be placed at the back of the next lower level queue. The job is next serviced when it reaches the head of that queue, if the first queue is empty. As long as the job continues using the full quantum provided at each level, it continues to move to the next lower-level queue. In general, this system is defined by several parameters: the number of queues, the service policy of each queue, a method of determining when to upgrade (demote) a job to a higher (lower) priority queue, a method of determining which queue a job will enter when it needs service. Finally, although a multi-level feedback queue is the most general scheme, it is also the most complicated one. It can be configured to match a specific system under design.

2. Static vs. Dynamic Scheduling:

There are two primary approaches to parallel/distributed processing and scheduling of tasks on a multiple processors/computers systems:

i. Static Scheduling which assumes that all parallel tasks in a computation are known apriori (that is, complete knowledge of tasks' characteristics at the system initiation time is required). They are statically mapped onto processors before the computation is initiated, and remain there throughout the entire computation. The mapping can be done off-line on an auxiliary scheduling processor, eliminating many of the information acquisition problems faced when scheduling. In such a paradigm, two different software design styles are used:

a) A universal task design style, as with the uniform grid used for solving partial differential equations, where each of the grid blocks is assigned to a task (Ortega & Voigt [85]); and

b) A network of communicating tasks, where the topology can be arbitrary and irregular and both the amount and frequency of intertask data transfer may vary during the computation; however, the number of tasks and their potential communication patterns are known. This permits a static mapping of tasks onto processors (Hoare [78]).

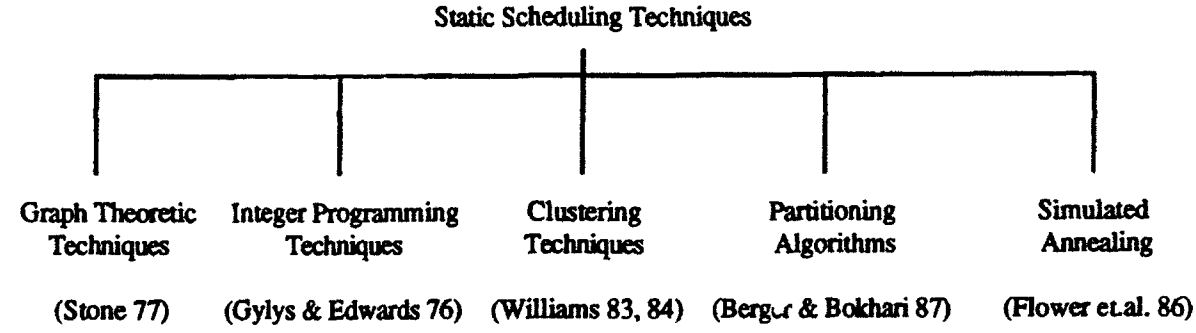


Fig. 2

Static algorithms have been subject of extensive study. They are relatively easy to design and have proven more tractable than dynamic algorithms. Their main disadvantages, however, are that they are often inflexible and cannot adapt to the dynamics of the environment.

Many approaches to the solution of static scheduling problems have been proposed, some of which are shown in the figure above.

ii. Dynamic Scheduling which calls for tasks to be scheduled as they arrive. A parallel computation is defined in terms of a dynamically created task precedence graph. New tasks are initiated and existing tasks terminate as the computation unfolds, and the mapping of tasks onto processors is done dynamically. Dynamically created tasks must be assigned to processors in real time by a scheduling algorithm executing on the multicomputer system. This dynamic view of computation differs in several significant ways from the static view. For example, the workload is

allowed to vary over time, multiple tasks may become eligible for execution given the results from a single task, and tasks are dynamically mapped onto processors using only partial knowledge of the global system state.

Algorithms for dynamic task scheduling may fall in one of two main categories:

a) *Task Placement Algorithms* where tasks are assigned to processors before the tasks begin execution, and all tasks execute where placed even though moving a task later might reduce the load imbalance.

b) *Task Migration Algorithms* where tasks are also assigned to processors before they begin execution; however, tasks can move after their initial placement.

Whether placement or migration is superior depends on the structure of the parallel computation. Several approaches to the solution of the dynamic scheduling problem have been introduced, some of which can be seen in the figure below.

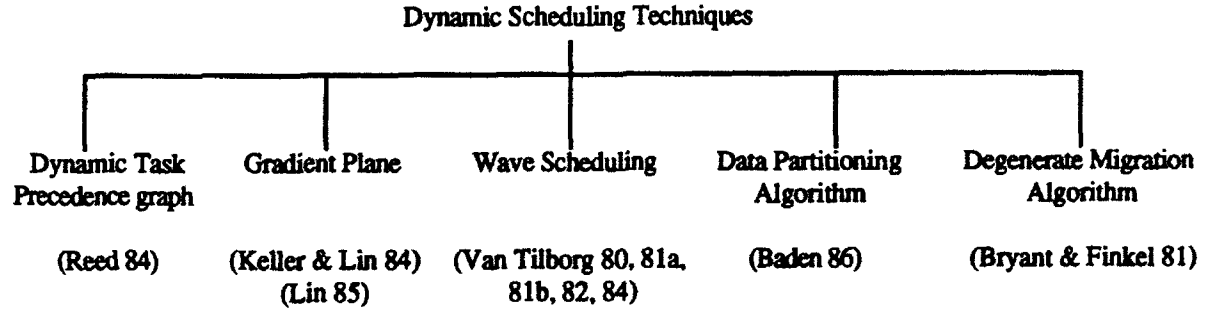


Fig. 3

Systems based on dynamic scheduling algorithms are flexible and can adapt to the dynamics of the environment easily.

E. Performance Measures

To compare the performance of one computer system to the performance of another or to some standard, a wide range of techniques is used depending on the particular situation. An important first step, however, is deciding exactly what aspects of the system's performance need to get measured and under what criteria. Performance measures that are in use may be grouped into two categories; job-oriented and system-oriented ones.

1. Job-Oriented Measures:

These measures express performance from the perspective of the tasks (or users). They are closely allied to the goal of user satisfaction. Examples of these measures are:

(i) **Flowtime**: This is the length of the time interval from submission to completion for a particular job. Flowtime is also frequently referred to as the *turnaround time* of the job. In most scheduling problems, the following relationship is assumed to hold

Turnaround time = Flowtime = Waiting time + Processing time

But if I/O time is considered in addition, then

Flowtime > Waiting time + Processing time

On the other hand, if the ready time (or release time) is zero, then, the turnaround time is equal to the completion time. Minimization of the average turnaround time (total flowtime) of jobs is frequently used as a performance criterion.

(ii) The **response time** of a system often serves as a performance measure, particularly in real-time (interactive) systems. It is usually defined as the time from job submission to the beginning of the first output produced by the job. Clearly, response time is related to user satisfaction and/or the usefulness of the system.

(iii) The **waiting time** is another measure used. It is the amount of time a job spends waiting in the system. More precisely, it is the amount of time, during the interval from arrival to completion of the job, that the job is ready for processing but not been processed.

(iv) The **makespan** is the total time required to complete a given set of jobs. Makespan is a fairly common performance measure. However, it does not relate to the satisfaction of an individual user in terms of the time required for his job to be completed.

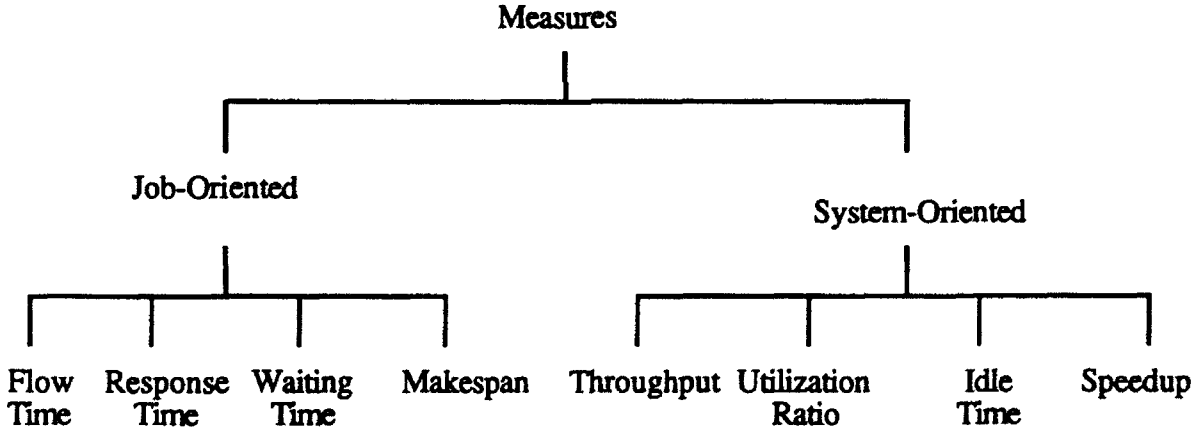


Fig. 4 Some Performance Measures For Computing Systems

2. System-Oriented Measures:

These are measures that take into account the system's point of view rather than that of the task or user. Several such measures exist, such as:

(i) The system's **throughput** which is the number of jobs that are completed per unit time. It is a measure of how much work a computing system is performing in a given time slot. It is important, when using this measure to compare algorithms, to perform tests with the same or very similar sets of jobs. Otherwise, throughput can be a misleading measure.

(ii) The system's **utilization ratio** which is normally expressed as the fraction (or percent) of the

total time that each processor is kept busy (or the average over the processors of the system). Thus, utilization can be expressed as the ratio of processing time to available time for each processor (or for the overall system). Essentially, the idea is to keep each processor as busy as possible. In doing so, we must avoid putting heavy load on some processors and light load on others, i.e. imbalance in load distribution. **Load balancing** (and sharing) techniques are used to equalize the utilization of the various processors in the system. These techniques are sometimes treated as scheduling strategies, with the purpose of improving the utilization of the system's resources.

(iii) The **idle time** of a processor is the time interval when the processor is available for processing, but not being used. Thus, the idle time of machine M_i, I_i, is given by

$$I_i = C_{max} - \sum_{j=1}^{n} P_{ij},$$

where C_{max} is the makespan and the summation represents the total processing time on M_i. The average idle time of the overall system is often considered. We may seek to minimize the total idle time, the weighted sum of idle time, or the mean idle time of the system. This measure is closely related to the previous one.

(iv) Another performance measure, which is used to indicate the improvement achieved in a system due to a specific procedure or due to a variation in the system itself, is the **speedup**. Speedup can be affected by several factors, such as the number of processors, the type of jobs, the interconnection between processors, the scheduling policy used, etc.. Speedup is usually evaluated for one of these parameters at a time, keeping the others constant. When the same set of jobs is executed on a single- and a multiple- processor system, an appropriate definition of speedup may be

$$S = \frac{\text{Sequential Processing time}}{\text{Concurrent Processing time}}.$$

F. Optimality Criteria and Strategies

1. General Overview:

There are numerous, complex, and often conflicting objectives that are to be achieved in performance evaluation studies. Many criteria have been suggested for comparing systems and algorithms and judging their effectiveness. Which characteristics are used for comparison can make a substantial difference in the determination of the best system or algorithm.

In general, let X_j be any quantity associated with job J_j. Then, one of the following measures might be of interest:

$$X = \sum_{j=1}^{n} X_j \qquad \text{Sum of } X_j \text{ for all n jobs;}$$

$$\bar{X} = \frac{1}{n} \sum_{j=1}^{n} X_j$$ Average (mean) value over all jobs;

$$X_{max} = \max_{1 \le j \le n} \{X_j\}$$ Maximum (peak) value over all jobs;

$$X_w = \sum_{j=1}^{n} w_j X_j$$ Total weighted sum over all jobs, with w_j being the weighting factors usually summing to 1.

The quantities defined above are **simple** objective functions or criteria of performance. More **general** objectives are constructed with nonlinear penalty functions $f_j(X_j)$, such as

$$X = \sum_{j=1}^{n} f_j(X_j)$$ Total penalty

$$X_{max} = \max_{1 \le j \le n} \{f_j(X_j)\}$$ Maximum penalty

In any of the above cases, we have **single** objectives. It is also possible to extend this to **multiple** objectives schemes by:

a) forming a **composite** objective such as Y + Z, with Y and Z being two single objectives; or

b) using **primary** and **secondary** objectives; for instance, by minimizing Y subject to $Z = Z^*$, where Z is the primary objective and Z^* is some optimized value while Y is the secondary objective.

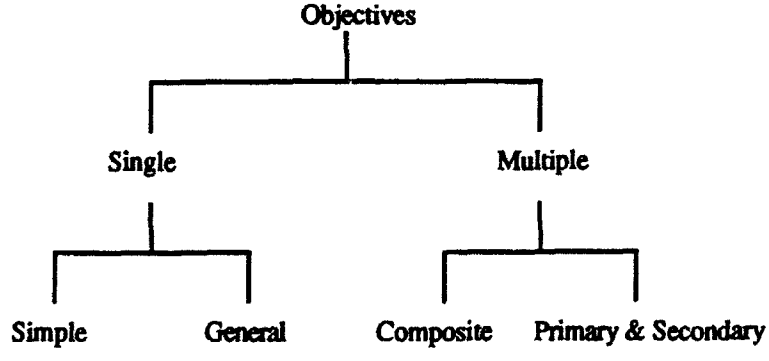


Fig. 5 Performance Evaluation Objectives

A policy is said to be **optimal** with respect to certain performance measure if it belongs to an equivalence class such that no nonempty classes which are preferred over this class exist.

If the performance metrics are regarded as cost (or reward) functions, then the optimality criterion might be:

(1) to minimize the maximum cost (*minimax*) - for instance, to minimize the maximum completion time: $C_{max} = \max_{1 \le j \le n} \{C_j\}$

(2) to maximize the average reward, such as, to maximize the average number of jobs being processed at a time t.

(3) to minimize the total cost (*minisum*), as in the case of minimizing the total completion time:

$$C = \sum_{j=1}^{n} C_j$$

In a given application, once the criteria have been defined, it is possible to evaluate the various algorithms under consideration. Table 1 shows some of the quantities of interest.

2. A Note On Relations Between Criteria:

Some criteria happen to be equivalent in the sense that a policy that is optimal with respect to one of them is also optimal with respect to the other(s). Knowing these equivalence relations reduces the number of separate problems that we have to deal with. Let us consider some of these cases. In section B we defined several time quantities related to the description of a given job J_j (See also fig. 1). Based on that information, the following relationships hold for each job:

$$L_j = F_j - a_j = C_j - r_j - a_j = C_j - d_j \qquad \text{(by definition)}$$

For n jobs:

$$\Sigma L_j = \Sigma F_j - \Sigma a_j = \Sigma C_j - \Sigma r_j - \Sigma a_j = \Sigma C_j - \Sigma d_j$$

Hence, $\overline{L} = \overline{F} - \overline{a} = \overline{C} - \overline{r} - \overline{a} = \overline{C} - \overline{d}$

Now, if $\overline{a}$, $\overline{r}$, $\overline{d}$ are given constants and a policy is optimal with respect to $\overline{F}$, then it will also be optimal for $\overline{C}$ and $\overline{L}$.

If $r_j = 0$ for all jobs, then $C_j = F_j$, and C_{max} and F_{max} are identical. However, F_{max} and C_{max} are quite distinct performance measures while $\overline{C}$ and $\overline{F}$ are essentially equivalent.

The average waiting time is also related to these measures since

$$C_j = r_j + W_j + \sum_{i=1}^{m} p_{ij} = r_j + F_j = L_j + d_j,$$

Therefore,

$$\overline{C} = \overline{r} + \overline{W} + \frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij} = \overline{r} + \overline{F} = \overline{L} + \overline{d}$$

Notice that $\overline{r}$, $\overline{d}$, $\overline{P}$ are constants for a given problem and independent of the policy. Thus, we minimize $\overline{C}$, $\overline{W}$, and $\overline{L}$ by minimizing $\overline{F}$, and the four measures are equivalent. The same can be said about C, F, W, and L. We should note that there is no parallel result concerning C_{max}, F_{max}, L_{max}, and W_{max}. Of course, there are special cases where two of these measures are equivalent (for instance, $r_j = 0 \Rightarrow C_{max} \equiv F_{max}$; and $d_j = D \Rightarrow C_{max} \equiv L_{max}$) but in general they are not.

If we minimize the final completion time of all the jobs, C_{max}, then the average number of processors being used at any one time is maximized and the average idle time of a processor is minimized, i.e. C_{max}, $\overline{N_p}$, $\overline{I}$ are equivalent measures.

We conclude from the above discussion that the following criteria serve to represent all **regular measures** introduced so far:

$$C,\ C_{max},\ C_w,\ L_{max},\ T,\ T_w$$

By a regular measure we mean a value that can be expressed as a nondecreasing function in the completion times of the job, i.e. a regular measure R is a real function of $C_1, C_2, ..., C_n$:
$R = f(C_1, C_2, ..., C_n)$ such that $C_1 \le C_1', ..., C_n \le C_n'$ implies that $R \le R' = f(C_1', C_2', ..., C_n')$.
C_{max}, $\overline{C}$, $\overline{F}$, F_{max}, $\overline{L}$, L_{max}, $\overline{T}$, T_{max}, U are regular measures. However, $\overline{E}$ and E_{max} are not regular measures.

III. CONCLUSION

In this report we have presented a summarized discussion of a number of significant factors that will have to be considered in any study involving performance evaluation of distributed computing systems. We also have identified some of the major problems which influence the performance of such systems. These factors and problems are introduced under six fundamental areas. These areas are: system's architecture and environment, application used as defined in terms of workload specification and characterization, types of constraints that exist in the system's hardware or software, types of service policies that may be used at a computing facility, some typical performance metrics, and finally, optimality criteria and the relations between some of them.

Clearly, much more research is needed to make use of the ideas introduced here. In any field, the identification and systematic classification of key ideas and issues are essential tasks for progress. Based on the issues and problems presented in this report, it is hoped that rigorous taxonomies can be developed to help determine standard terminology for performance evaluation of distributed computing systems. This rigorous taxonomy should be both *exhaustive* of the categories and *exclusive* with no overlap of the classification criteria defined. This in turn, will provide an unambiguous categorization for every concept presented to the taxonomy.

ACKNOWLEDGEMENTS

The author wishes to thank Richard Metzger, Mary Denz, Tom Lawrence, and Jerry Dussault of Rome Laboratories, Dr. Rex Gantenbein of the University of Wyoming, and Dr. Gary Craig of Syracuse University for their support and helpful discussions and ideas. This work was supported by the AFOSR Summer Research Program. The views and opinions expressed in this report are those of the author and should not be construed as an official Department of Defense position.

REFERENCES

Anderson, G. A. and Jensen, E. D., (1975); "Computer Interconnection Structures:Taxonomy, Characteristics, and Examples," Computing Surveys, Vol. 7, No. 4, pp. 197-213.

Baden, S. B., (1986); "Dynamic Load Balancing of a Vortex Calculation Running on Multiprocessors," Lawrence Berkeley Laboratory, Univ. of California, Berkeley, CA.

Berger, M. J., and S. H. Bokhari, (1987); "A Partitioning Strategy for Nonuniform Problems on Multiprocessors," IEEE Trans. on Computers, Vol. C-36, No. 5, pp. 570-580.

Bryant, R. M. and R. A. Finkel (1981); "A Stable Distributed Scheduling Algorithm," *Proc. of the Second International Conf. on Distributed Computing Systems*, April 1981.

Duncan, R. , (1990); "A Survey of Parallel Computer Architectures," Computer Magazine, Vol. 23, No. 2, pp. 5-16.

Dunnigan, T.H., (1987); "Hypercube Performance," in M.T. Heath (ed.), HYPERCUBE MULTIPROCESSORS 1987, SIAM, Philadelphia, pp. 178-191.

Enslow, P. H., Jr. , (1977); "Multiprocessor Organization-A Survey," Computing Surveys, Vol. 9, No. 1, pp. 103-129.

Flower, J. W., S. W. Otto and M. C. Salama, (1986); "A Preprocessor for Irregular Finite Element Problems," CalTech Concurrent Computation Project Memo #292, July 1986.*

Flynn, M. J., (1972); "Some Computer Organizations and Their Effectiveness," IEEE Trans. on Computers, Vol. C-21, No. 9, pp. 948-960, Sept. 1972.

Gehringer, E. F., D. P. Siewiorek and Z. Segall, (1987); PARALLEL PROCESSING: THE CM* EXPERIENCE, Digital Press.

Gylys, V. B., and J. A. Edwards, (1976); "Optimal Partitioning of Workload for Distributed Systems," *Proc. of the fall IEEE COMPCON*, pp. 353-357, Sept. 1976.

Heath, M. T. (1987); "Hypercube Applications at Oak Ridge National Laboratory," in M.T. Heath (ed.), HYPERCUBE MULTIPROCESSORS - 1987, SIAM, Philadelphia.

Ho, C. and S. Johnson, (1986); "Distributed Routing Algorithms for Broadcasting and Personalized Communication in Hypercubes," *Proc. of 1986 Interl. conf. on Parallel Processing*, Aug. 1986, pp. 640-648.

Hoare, C. A. R., (1978); "Communicating Sequential Processes," Comm. ACM, Vol. 21, No. 8, pp. 666-677, Aug. 1978.

Jones, A. K., and P. Schwarz,(1980); "Experiences Using Multiprocessor Systems--A Status Report," Computing Surveys, Vol. 12, No. 2, pp. 121-165, June 1980.

Keller, R. M., and F. C. H. Lin, (1984); "Simulated Performance of a Reduction-based Multiprocessor," Computer Mag., Vol. 17, No. 7, pp. 70-82.

Lan, Y., A. H. Esfahanian, and L. M. Ni, (1988a); "Distributed Multi-destination Routing in Hypercube Multiprocessors," *Proc. of the Third Conference on Hypercube Concurrent Computers and Applications*, Pasadena, CA, Jan. 1988.

Lan, Y., A. H. Esfahanian, and L. M. Ni, (1988b); "Multicast in Hypercube Multiprocessors," *Proc. of the 1988 Phoenix Conf. on Computers and Communications*, March 1988, pp. 27-30.

Lin, F. C. H., (1985); "Load Balancing and Fault Tolerance in Applicative Systems," Ph.D. Dissertation, Dept. of Computer Science, Univ. of Utah.

Ortega, J., and R.G. Voight, (1985); "Solution of Partial Differential Equations on Vector and Parallel Computers," SIAM Review, Vol. 27, No. 2, pp. 149-240, June 1985.

Reed, D. A., (1984); "The Performance of Multimicrocomputer Networks Supporting Dynamic Workloads," IEEE Trans. on Computers, Vol. C-33, No. 11, pp. 1045-048.

Shih, Y., and J. Fier, (1987); "Hypercube Systems and Key Applications," *Chapter 6* in PARALLEL PROCESSING FOR SUPERCOMPUTING AND AI, K. Hwang, and D. DeGroot, (eds.), McGraw-Hill, New York.

Stone, H. S., (1977); "Multiprocessor Scheduling with the Aid of Network Flow Algorithms," IEEE Trans. on Software Engineering, Vol. SE-3, No. 1, pp. 85-93, Jan. 1977.

Van Tilborg, A. M., (1982); "Network Computer Operating Systems and Task Force Scheduling," Ph.D. Dissertation, Dept. of Computer Science, SUNY-Buffalo.*

Van Tilborg, A. M., and L. D. Wittie (1984); "Wave Scheduling-Decentralized Scheduling of Task Forces in Multicomputers," IEEE Trans. on Computers, Vol. C-33, No. 9, pp. 835-844 .

Van Tilborg, A. M., and L. D. Wittie, (1980); "High-level Operating Systems Formulation in Network Computers," *Proc. of the 1980 International Conf. on Parallel Processing*, August 1980, pp. 131-132.

Van Tilborg, A. M., and L. D. Wittie, (1981a); "Wave Scheduling: Distributed Allocation of Task Forces in Network Computers," *Proc. of the Second International Conf. on Distributed Computer Systems*, pp. .

Van Tilborg, A. M., and L. D. Wittie, (1981b); "Distributed Task Force Scheduling in Multimicrocomputer Networks," *Proc. of the National Computer Conf.*, Vol. 46, pp. 283-289.

Williams, E. A., (1983); "Assigning Processes to Processors in Distributed Systems," *Proc. of the 1983 Intern. conf. on Parallel Processing*, August 1983, pp. 404-406.

Williams, E. A., (1984); "The Effect of Queueing Disciplines on Response Times in Distributed Systems," *Proc. of the 1984 Intern. conf. on Parallel Processing*, August 1984, pp. 330-332.

Quantity	The Value	Description
C	$\sum_{j=1}^{n} C_j$	Total completion time
$\overline{C}$	$\frac{1}{n}\sum_{j=1}^{n} C_j$	Mean completion time
C_w	$\sum_{j=1}^{n} w_j C_j$	Total weighted completion time
C_{max}	$\max_{1\le j\le n}\{C_j\}$	Maximum Completion time (makespan)
T	$\sum_{j=1}^{n} T_j$	Total tardiness
$\overline{T}$	$\frac{1}{n}\sum_{j=1}^{n} T_j$	Mean tardiness
T_w	$\sum_{j=1}^{n} w_j T_j$	Total weighted tardiness
T_{max}	$\max_{1\le j\le n}\{T_j\}$	Maximum tardiness
$\overline{L}$	$\frac{1}{n}\sum_{j=1}^{n} L_j$	Mean Lateness
$\overline{U}$	$\sum_{j=1}^{n} U_j$	Number of tardy jobs
U_w	$\sum_{j=1}^{n} w_j U_j$	Weighted number of tardy jobs
$\overline{N}_u$	$\frac{1}{C_{max}}\int_0^{C_{max}} N_u(t)\,dt$	Average number of jobs still to be completed by time t over the time period

Table 1 Some Relevant Optimality Parameters

THE EFFECTS OF ARRAY BANDWIDTH ON PULSE RADAR PERFORMANCE AND TIME-DELAYED SUBARRAY COMPENSATION

Charles T. Widener
Department of Electrical and Computer Engineering

Syracuse University
Syracuse, NY 13244

Final Report for:
Summer Research Program
Rome Laboratory

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D.C.

July 1992

THE EFFECTS OF ARRAY BANDWIDTH ON PULSE RADAR PERFORMANCE AND TIME-DELAYED SUBARRAY COMPENSATION

Charles T. Widener
Department of Electrical and Computer Engineering
Syracuse University

ABSTRACT

The basic principles governing phase scanning of array antennas are briefly outlined. Examination of these principles shows that a frequency dependence in the phase steering equation leads to an inability of a phase scanned array to radiate all frequencies in a single direction. Using this fact the concept of array bandwidth is introduced and an appropriate definition given. By adopting a linear system representation for the antenna, it is illustrated how the resulting loss of signal energy at the receiver can be conveniently calculated. Having defined the problem and its major effect on radar performance, the compensation technique of time-delay subarraying is discussed. Special consideration is given to systems employing linear frequency modulation pulse compression. Plots of the number of subarrays required to maintain a certain level of radar performance vs maximum scan angle are given for various system parameters.

THE EFFECTS OF ARRAY BANDWIDTH ON PULSE RADAR PERFORMANCE AND TIME-DELAYED SUBARRAY COMPENSATION

Charles T. Widener

I. Introduction

It is well known in the theory of array antennas that the main beam of the antenna can be steered in space (scanned) by the application of a linear phase progression across the antenna aperture. At a frequency f and wavelength λ, the phase required on an element a distance x from the antenna center to steer the beam to an angle θ from broadside is

$$\phi(x,\theta) = \frac{2\pi}{\lambda} x \sin\theta = \frac{2\pi f}{c} x \sin\theta \qquad (1)$$

where c is the velocity of propagation. For elements arranged in a line, with uniform separation d, the phase difference from one element to another is given by

$$\Delta\phi = \frac{2\pi}{\lambda} d \sin\theta = \frac{2\pi f}{c} d \sin\theta \qquad (2)$$

The angle θ to which the beam points is changed by adjusting the value of ϕ on each element. If the applied phase ϕ on each element is allowed only values between 0 and 2π then the array is said to be phase scanned. Devices used to provide this kind of phasing are called phase shifters. Phase scanned arrays are frequency sensitive, as may be seen from an inspection of eq. (1) or (2) as follows. If the phase ϕ on each element is held constant and has been adjusted to provide a scan angle θ for a given operating frequency f, then since ϕ is constant so is the quantity f sin θ. If f sin θ equals a constant quantity, then any change in f causes a change in sin θ such that

their product remains unchanged. The resulting effect is that for phase scanned arrays, a change in frequency causes a shift in scan angle.

Time-delaying is another method that can be used to scan an array. Time-delay scanning uses delay lines instead of phase shifters to provide the necessary phasing on each element. A delay line in its simplest form is just a piece of transmission line. The length of the delay line is adjusted to provide a time delay of $t=(d/c)\sin\theta$ (see Fig. 1) from element to element. With time-delay scanning the scan angle is independent of frequency.

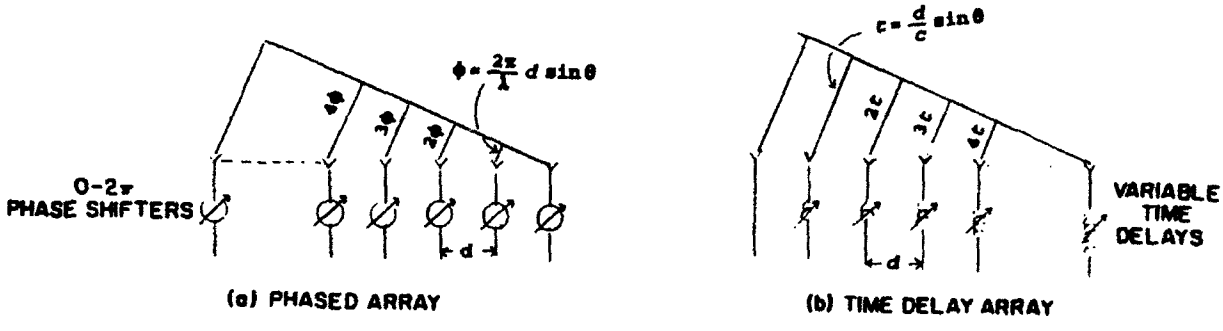


Fig. 1 Phase relationships for Beam Steering

A significant distinction between phase shift and time delay scanning can be seen by noting that phase shifters provide a total phase shift of $0 \le \phi < 2\pi$ and delay lines provide a phase shift of

$$\phi = 2\pi \frac{l}{\lambda} \qquad (3)$$

for a transmission line of length l. If l is many times λ, then the resultant phase is many multiples of 2π. This will always be the case for antennas of practical size and beamwidth.

The phase relationships given by eqs. (1) and (2) are derived by assuming each radiating element of the array operates in a single frequency CW mode. Most radars, however, operate in a pulsed mode rather than CW. The frequency content of a pulsed waveform consists of a spectrum which extends over a frequency band Δf_p (the subscript denotes pulse) centered around the transmitted frequency f_0. When an array antenna is *phase scanned* to an angle θ_0, corresponding to the frequency f_0, the frequency dependence of ϕ will cause frequencies other than f_0 to scan to angles other than θ_0. The amount of angular deviation $\Delta\theta$ from θ_0 for a frequency of $f_0+\Delta f$ is given by

$$\Delta\theta \approx -\frac{\Delta f}{f_o} \tan\theta_o \qquad (4)$$

This is known as the aperture effect. A similar phenomenon occurs when each element of the array is fed by a transmission line of differing length, such as in series fed arrays. In this case, the phase of excitation at each element will vary according to the length of the line and frequency used (exactly like time-delay scanning). The added phase variation causes a further deviation in scan angle. This is known as a feed effect. The two effects are additive and can be understood independently of one another. For the remainder of this report it will be assumed that all radiating elements are fed in parallel by equal line lengths (also known as corporate fed) so

that feed effects can be ignored.

The definition of array bandwidth for pulsed waveforms, as proposed by Frank [1], is that range of frequencies, Δf_a, (the subscript a denoting array) which causes the angular deviation $\Delta\theta$ to lie within the half power beamwidth θ_{HPBW} defined for CW operation on the same array. This report considers the limitations on the ability of a phase scanned array to transmit (and receive) a pulse spectrum due to the array bandwidth effect and the resulting effects on radar performance.

II. Array Bandwidth Limitations

One obvious consequence of the angular dispersion described by equation (4) is that some of the energy intended to radiated in a direction θ will go elsewhere. The result is a loss of energy on target. This loss is sometimes referred to as one-way loss since only the transmit portion of the round trip is considered.

The loss of energy on target can be calculated by considering the scanned array as a frequency dependent network, which has an additional dependence on θ. The input to the array is the transmitted pulse, denoted $s(t)$ in the time domain. The output of the array $y(t,\theta)$ will be the convolution of the input signal with the impulse response of the array $a(t,\theta)$ given by the expression:

$$y(t,\theta) = \int_{-\infty}^{\infty} s(\tau)\, a(t-\tau,\theta)\, d\tau \qquad (5)$$

Since convolution in the time domain implies multiplication in the frequency domain, the output spectrum $Y(\omega,\theta)$ takes the form

$$Y(\omega,\theta) = S(\omega)A(\omega,\theta) \tag{6}$$

The energy of the output waveform at a scan angle θ over the array bandwidth Δf_a defined above is given by the expression

$$E(\theta) = \int_{f_0-\frac{\Delta f}{2}}^{f_0+\frac{\Delta f}{2}} |S(\omega)A(\omega,\theta)|^2 \, d\omega \tag{7}$$

The loss in energy due to array bandwidth limitations can be expressed as the ratio of the energy on target at a scan angle θ to the energy on target for a scan angle $\theta=0^\circ$,

$$Loss\ ratio = \frac{\int |S(\omega)A(\omega,\theta)|^2 \, d\omega}{\int |S(\omega)|^2 \, d\omega} \tag{8}$$

This loss has been computed by several authors, notably Frank [1], Hammer [2], and Adams [3].

An equivalent way of understanding the loss mechanism for broadband phased-arrays is in terms of the aperture fill time. As shown in Fig. 2, if a signal is incident from θ_0, the pulse front reaches one end of the array before the other. The signal travels a distance L $\sin\theta_0$ farther to the last element than to the first. The time it takes for the signal to be present in all elements of the array is the aperture fill time $T=(L/c)\sin\theta_0$. It is readily shown that the array bandwidth can be expressed in terms of the aperture fill time as $\Delta f_a \approx 1/T$.

It should be clear that if the array bandwidth is larger than the bandwidth of the pulse waveform Δf_p, at a particular scan angle, then the energy loss will be small. As the array bandwidth narrows due to scanning,

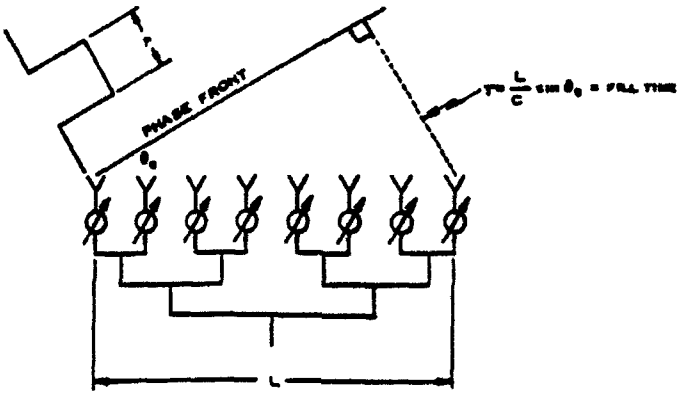


Fig. 2 Aperture Fill Time

more of the pulse energy is lost. Since pulse length τ is related to the pulse bandwidth Δf_p by the relation $\tau \approx 1/\Delta f_p$, the ratio T/τ can be used as a reference for energy loss. Figure 3 shows a plot of loss in gain due to one

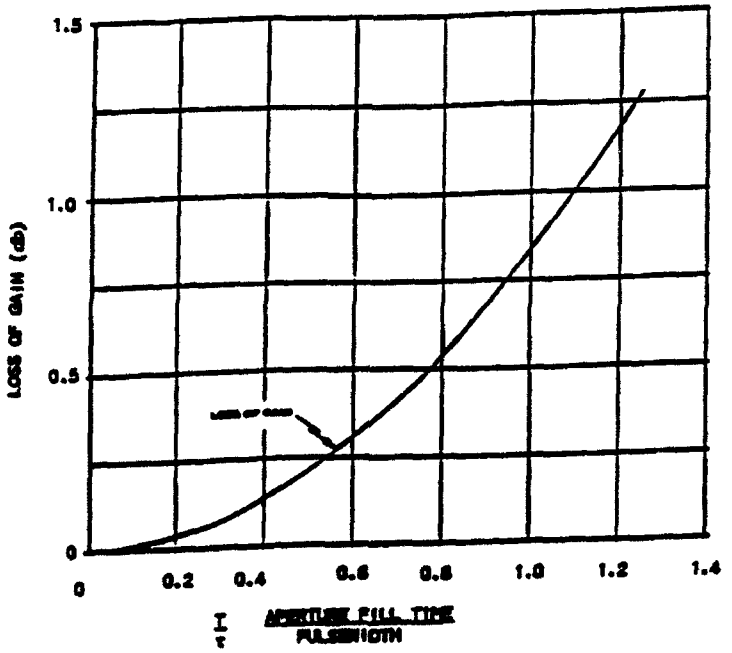


Fig 3. Loss of Gain due to Scanning (after Frank)

way loss of energy as a function of the ratio T/τ for a rectangular pulse spectrum, assuming a uniformly illuminated aperture (after Frank).

III. System Degradation

While the loss of energy on target for a phase scanned array can account partially for radar performance degradation, it is more meaningful to consider the entire signal round trip, i.e., both transmit and receive. In the transmit mode, each array element is excited at the same time (not true for a time-delay scanned array). If a target P is located at some angle θ_0 off broadside, then the signals from each element will arrive at the target slightly displaced in time by

$$t_d = \frac{d}{c} \sin\theta_0 \qquad (9)$$

Each elemental signal will be reflected and returned to the array and received by all the elements, suffering the same time delay effect on receive as on transmit. Two effects are immediately clear:

1) the received pulse is smeared in time (spreading occurs) and

2) the amplitude of the received signal is distorted compared to the transmitted signal.

The result of these effects is a loss of signal energy at the receiver. This loss is sometimes called a two way loss because it includes losses incurred in the signal round-trip. Calculation of the two way loss is again treated conveniently in the frequency domain as a problem in linear systems. The spectrum of the transmitter output pulse is first modified (or filtered) by the transmission transfer function of the antenna array. The

resulting transmitted spectrum is then reflected off a target and upon reception, is modified by the receive transfer function of the antenna array.

*Note: The transmit and receive patterns of the array may employ different aperture weightings to maximize certain features of each. For example, uniform weighting on transmit allows maximum energy on target, while a tapered weighting on receive may be used to maintain sidelobes at some predetermined low level.

The resulting spectrum incident upon the receiver would be

$$Y(\omega,\theta_0) = S(\omega)\,A_T(\omega,\theta_0)\,A_R(\omega,\theta_0) \qquad (10)$$

where A_T and A_R are respectively the transmit and receive transfer functions of the antenna array. An appropriate matched filter would have as its transfer function $[S(\omega)A_T(\omega,\theta_0)A_R(\omega,\theta_0)]^*$, where * denotes complex conjugate. This particular implementation is dependent on scan angle θ and would be difficult to achieve in practice. A more likely, simpler receiver would use a filter matched to $S(\omega)$ alone. Signal-to-noise ratio (SNR) loss curves for both cases (receiver matched to $S(\omega)$, and matched to $Y(\omega,\theta_0)^*$) have been presented in an excellent paper by J. R. Sklar [4] for a uniformly weighted aperture, and by Rothenberg and Schwartzman [5] for two Taylor weighted apertures. A plot of SNR loss for the uniformly weighted case is shown in Figure 4 (after Sklar). A simple closed form expression describing the SNR loss due to array bandwidth limitations is given by Barton [6] as

$$L_x = 1 + 2\frac{B_3^2}{B_a^2} \qquad (11)$$

where: L_x is the loss factor

B_a is the half-power bandwidth of the array

and B_3 is the width of the receiver output filter (assumed narrowband).

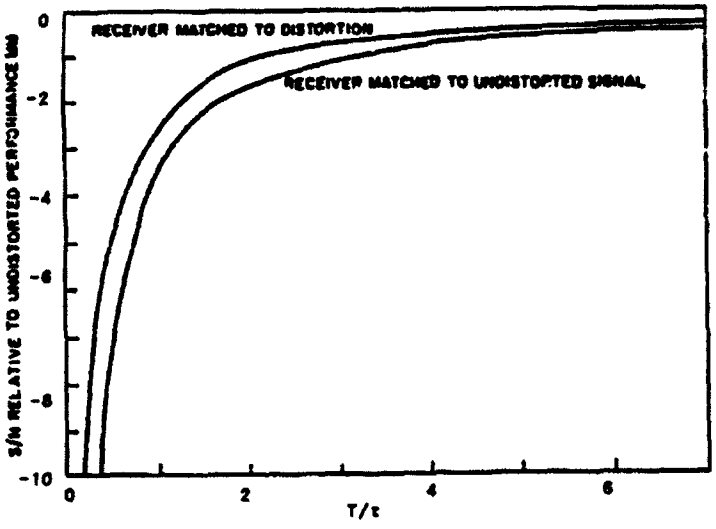


Fig. 4 SNR loss vs. T/τ (after Sklar)

Additional performance degradations, also discussed by Sklar, are loss of range resolution and loss of range accuracy. These degradations are mainly the result of received signal spreading in time.

IV. Time-Delayed Subarrays

As component technology and radar theory becomes more and more advanced, the trend has been to build radars with higher resolution capabilities in both range and azimuth (cross-range). Requirements for fine range resolution include the use of a short pulse (wide bandwidth); similarly, as more resolution in azimuth is required, it is necessary to use narrow beamwidths,

which implies array dimensions much, much greater than the wavelength ($L \gg \lambda$). Unfortunately, array bandwidth effects become more pronounced when a short pulse is used with a large array.

One method of overcoming performance degradation due to array bandwidth effects is through the use of time-delayed subarrays. Time-delaying, already mentioned as one form of scanning, is currently prohibitively costly and difficult to implement on a per element basis. But it can be applied to small portions of the overall antenna (called subarrays) as an effective way to increase array bandwidth, and thereby decrease the SNR loss.

An easy way to understand the increase in bandwidth using time-delayed subarrays is to note that signal returns for each subarray are in time coincidence, the subarrays having been essentially "stacked" in time. Computation of array bandwidth is then done using the length of a single subarray section, L_a. Since the bandwidth of a subarray, is proportional to λ/L_a, the shorter the length of the subarray, the wider the bandwidth becomes. Comparison of subarray bandwidth to array bandwidth indicates a factor of N increase when N subarrays are used. This provides an excellent means of decreasing the SNR degradation that accompanies phase scanned arrays when used with wideband pulses.

When it is necessary to employ subarraying to avoid excessive SNR loss, the primary factors for consideration are:

a) pulsewidth τ

b) overall antenna length L

c) maximum scan angle θ_0

d) maximum SNR degradation which can be tolerated, and

e) antenna aperture weighting taper(s) employed.

The pulsewidth τ and antenna length L can be conveniently expressed together as the ratio τ/T_0, where T_0 is the transit time of L by the speed of light, i.e. $T_0=L/c$.

It is clear from Fig. 4 (for the case of uniform weightings) that when T/τ is greater than 3, the resulting SNR loss is less than 1 dB. Using 1 dB as the maximum tolerable loss in SNR, the minimum number of subarrays required is graphically represented in Figure 5 as a function of maximum scan angle.

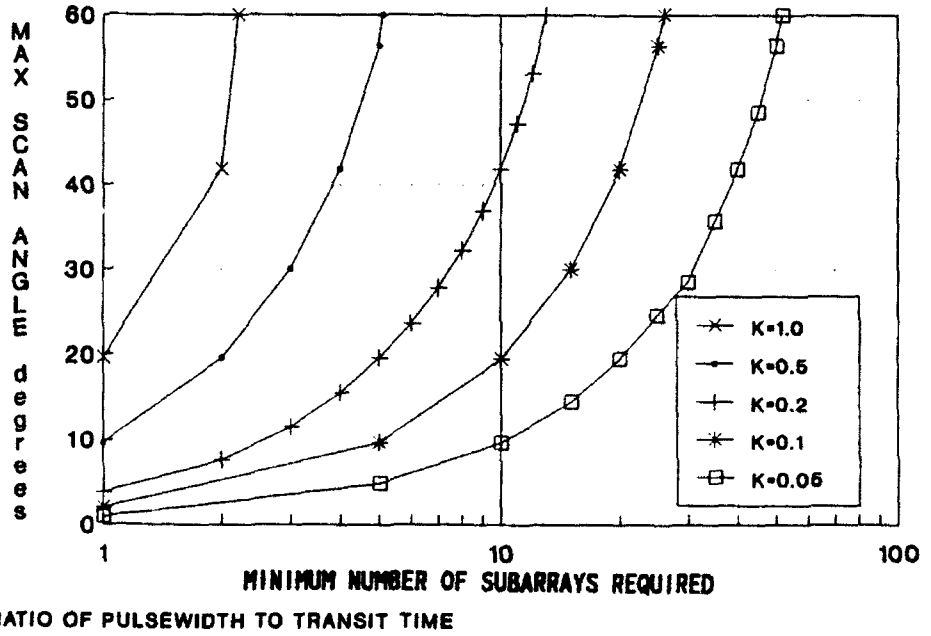


Fig. 5 Minimum Number of Subarrays Required for 1 dB or less SNR loss vs Max Scan Angle for τ/T_0= 1.0, 0.5, 0.2, 0.1, 0.05

The curves are plotted for various ratios of τ/T_0 when the receiver is

matched to the antenna input pulse spectrum $S(\omega)$. If the tolerable SNR loss were more (or less) than 1 dB, similar charts could be made to conform to that criteria.

V. Pulse Compression Considerations

The effects of array bandwidth limitations on radar performance have been presented for the special case of single frequency, pulse mode operation. While pulsed radars are typical in modern day use, modes other than single frequency are very common as well. It was pointed out in section IV that the trend today is towards higher resolution capability. In this context it was mentioned the use of a short pulse is one method used to achieve this goal. However, as is usually the case, pushing one parameter to its limit generally reflects adversely in another parameter. Short pulse modes are no exception. While short pulse modes can increase the resolution in range they inherently put less energy on target which decreases the maximum detection range. Pulse compression is a technique that employs a modulated transmitter waveform that maintains a high level of energy on target while achieving the high range resolution associated with a short pulse. Two transmitter modulations commonly used are linear frequency modulation (LFM) and phase coding. Both of these modulations will interact with the scanning mechanisms of phase scanned arrays. An analysis by J. B. Payne [7] indicates that a phase scanned antenna's response to a pulse compression waveform is identical to its response to a pulse equal in width to that of the compressed pulsewidth. One might assume that the curves presented in Figure 5 could be used equally as well for a LFM waveform with τ replaced by $(f_2-f_1)^{-1}$, where f_2 and f_1 are the

limits of the frequency modulation of the LFM waveform. Unfortunately this is too coarse an assumption.

Regarding Payne's analysis with respect to LFM pulse compression, it needs to be mentioned that his analysis assumes a matched filter receiver. It is well known [8-9] that the matched filter output to an LFM pulse has undesirable time sidelobes which can mask a weak target or be mistaken for targets themselves. These sidelobes are reduced by amplitude weighting the received signal spectrum within the filter, much as an antenna aperture is weighted to reduce sidelobes. The resulting loss due to filter mismatch is on the order of 1 to 2 dB for most practical weightings.

Knittel [10], however, has done a comprehensive study on the SNR loss and range resolution degradation resulting from array dispersion for systems using LFM pulse compression. His results are based on a comparison to an ideal system having a dispersionless array and a receiver weighting filter designed for 40 dB time sidelobes (Taylor n=8). His analysis shows that there is an optimum pulse compression bandwidth which minimizes both SNR loss and pulse shape distortion. Although as an example he has examined a system with specific parameters, he has generalized his results to include arbitrary aperture size D/λ, arbitrary scan angle θ, and arbitrary fractional signal bandwidth $B=(f_2-f_1)/f_0$, where f_0 is the midband frequency of the pulse compression modulation. It should also be mentioned that Knittel assumes a larger useable bandwidth for the array than that proposed by Frank.

Abstracting from Knittel's SNR loss curve for a parallel feed array with uniform weighting on transmit and Taylor 30 dB (n=6) weighting on receive, it is found that time-delay subarraying should be used if the ratio of compressed

pulsewidth to aperture fill time τ_c/T is less than 0.88, in order to maintain a SNR loss of less than 1 dB. Based on this result, the number of subarrays required to maintain less than 1 dB SNR degradation vs maximum scan angle for various values of $K=\tau_c/T_0$ is shown in Figure 6.

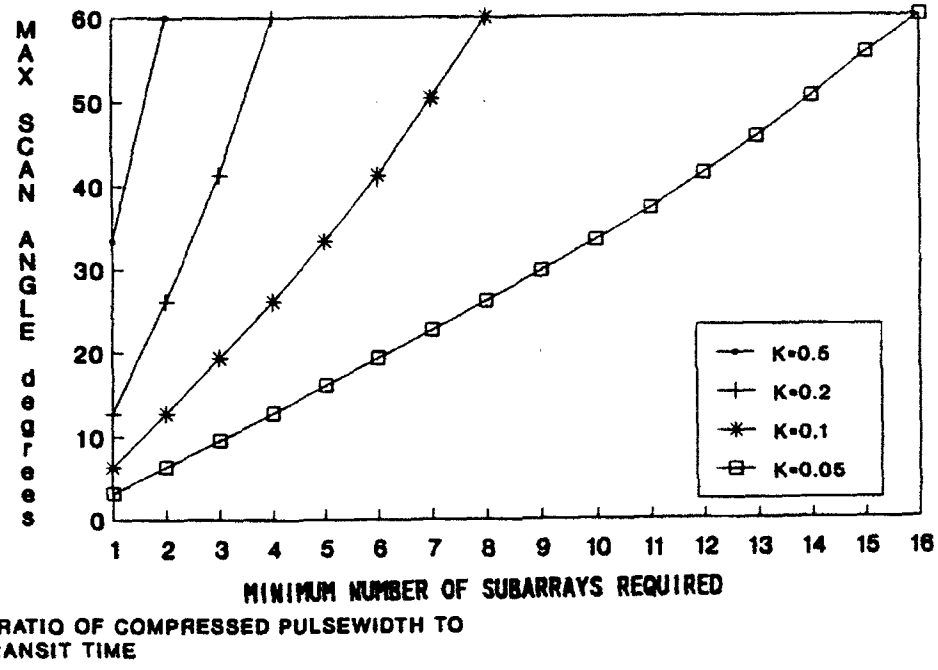


Fig. 6 Minimum Number of Subarrays Required for 1 dB or less SNR loss vs Max Scan Angle for τ_c/T_0= 0.5, 0.2, 0.1, 0.05

VI. Summary

In the preceding pages it has been shown that when the elements of an antenna array are controlled by phase shift devices to steer the antenna beam in space, the direction of the energy is dependent on the frequency of the transmitted RF signal. For the case of pulsed radars it was seen that due to the inherent nature of a pulsed waveform, some of the transmitted energy is

radiated in directions other than the intended scan angle. This leads to a definition of a bandwidth for the array which in essence describes a band of frequencies which will radiate in the intended direction. In sections II and III the resulting effects on radar performance due to array bandwidth limitations were discussed. The first effect noted was a loss of energy on target due to angular dispersion. The second, and more important effect, was the loss of signal energy at the receiver resulting in SNR degradation. Curves for both the one- and two-way loss were presented to illustrate their dependence on pulsewidth τ, antenna length L, and scan angle θ.

In Section IV, a compensation technique using time-delayed subarrays was discussed and plots given for the number of subarrays required to maintain less than 1 dB SNR degradation for various ratios of τ/T_0 as a function of maximum scan angle. Similar plots were presented in Section V for applications using LFM pulse compression.

As a final note, it is mentioned that the subarraying technique referred to in this report is the conventional or contiguous method. Other methods such as overlapped and interleaved can also be used to further reduce SNR loss and minimize grating lobe levels. Interested readers are referred to an overview of such techniques by R. Tang [11].

REFERENCES

[1] J. Frank, "Bandwidth criteria for phased array antennas," in *Phased Array Antennas*, Oliner and Knittel, Eds., Artech House, Inc., Mass., 1972, pp. 243-253

[2] I. W. Hammer, sec.13 in Radar Handbook, M. I. Skolnik, Ed., McGraw-Hill Book Company, New York, 1970

[3] W. B. Adams, "Phased-array radar performance with wideband signals," Supplement to IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-3, No. 6, Nov. 1967, pp. 257-271

[4] J. R. Sklar, "Short-pulse limitation of phased arrays," in "Phased Array Radar Studies, July 1,1960 to July 1, 1961," M.I.T. Lincoln Lab., Lexington, Mass., Tech. Rep. 236, part 3, ch. II, Nov. 13, 1961

[5] C. Rothenberg and L. Schwartzman, "Phased array signal bandwidth," 1969 G-AP Int. Symp. Digest, pp. 116-123

[6] D. K. Barton, *Modern Radar System Analysis*, Artech House, Inc., Mass., 1988, p. 187

[7] J. B. Payne III, "Relation between a generalized wideband signal and its equivalent short pulse when applied to array antennas," IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-2, No. 2, Mar. 66, pp. 233-234

[8] M. I. Skolnik, *Introduction to Radar Systems*, McGraw-Hill Book Company, New York, 1980, p. 426

[9] D. K. Barton, *Modern Radar System Analysis*, Artech House, Inc., Mass., 1988, pp. 223-224

[10] G. H. Knittel, "Relation of radar range resolution and signal-to-noise ratio to phased-array bandwidth," *IEEE Trans. Antennas Propagat.*, vol. AP-22, No. 3, May 1974 pp. 418-426

[11] R. Tang, "Survey of time-delay beam steering techniques," in Phased Array Antennas, Oliner and Knittel, Eds., Artech House Inc., Mass., 1972, pp. 254-260

METAMODEL APPLICATIONS USING TERSM

Michael A. Zeimer
Graduate Student

and

Dr. Jeffrey D. Tew
Assistant Professor

Department of Industrial and Systems Engineering
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

and

Dr. Robert G. Sargent
Professor

Simulation Research Group
L. C. Smith College of Engineering and Computer Science
Syracuse University
439 Link Hall
Syracuse, New York 13244

Final Report for:
Summer Research Program
Rome Laboratory

Sposored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D. C.

September 1992

METAMODEL APPLICATIONS USING TERSM

Michael A. Zeimer
Graduate Student

and

Dr. Jeffrey D. Tew
Assistant Professor

Department of Industrial and Systems Engineering
Virginia Polytechnic Institute and State University

and

Dr. Robert G. Sargent
Professor

Simulation Research Group
L. C. Smith College of Engineering and Computer Science
Syracuse University

Abstract

Tactical simulation models are often used to assess vulnerabilities and capabilities of combat systems and doctrines. Due to the complexity of tactical simulation models, it is often difficult to assess the relationship between input factors and the performance of the simulation model. To facilitate this type of assessment, simulation analysts often use the simulation model to emirically construct a *black-box* approximation of the causal and time dependent behavior of the simulation model. This type of approximation is known as a *metamodel* and can be viewed as a summary of the behavior of the simulation model. We demonstrate this technique in the context of an example using TERSM (Tactical Electronic Reconnaissance Simulation Model). The results indicate that metamodeling is applicable to tactical simulation models and that the technique has a wide range of uses.